Non-interactive CCA2-Secure Threshold Cryptosystems: Achieving Adaptive Security in the Standard Model Without Pairings

Julien Devevey[1(✉)], Benoît Libert[1,2], Khoa Nguyen[3], Thomas Peters[4], and Moti Yung[5]

[1] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, Inria, UCBL), Lyon, France
julien.devevey@ens-lyon.fr
[2] CNRS, Laboratoire LIP, Lyon, France
[3] SPMS, Nanyang Technological University, Singapore, Singapore
[4] FNRS & UCLouvain, ICTEAM, Louvain-la-Neuve, Belgium
[5] Google and Columbia University, New York, USA

Abstract. We consider threshold public-key encryption, where the decryption servers distributively hold the private key shares, and we need a threshold of these servers to decrypt the message (while the system remains secure when less than the threshold is corrupt). We investigate the notion of chosen-ciphertext secure threshold systems which has been historically hard to achieve. We further require the systems to be, both, adaptively secure (i.e., secure against a strong adversary making corruption decisions dynamically during the protocol), and non-interactive (i.e., where decryption servers do not interact amongst themselves but rather efficiently contribute, each, a single message). To date, only pairing-based implementations were known to achieve security in the standard security model without relaxation (i.e., without assuming the random oracle idealization) under the above stringent requirements. Here, we investigate how to achieve the above using other assumptions (in order to understand what other algebraic building blocks and mathematical assumptions are needed to extend the domain of encryption methods achieving the above). Specifically, we show realizations under the Decision Composite Residuosity (DCR) and Learning-With-Errors (LWE) assumptions.

Keywords: Threshold cryptography · Adaptive security · Non-interactive schemes · Standard model · Chosen-ciphertext security · DCR · LWE

1 Introduction

Threshold cryptography [17,36,38,39] avoids a single point of failure by splitting the secret key into $\ell > 1$ shares and handing them over to different servers.

J. A. Garay (Ed.): PKC 2021, LNCS 12710, pp. 659–690, 2021.
https://doi.org/10.1007/978-3-030-75245-3_24

This is done in such a way that any set of size at least $t \leq \ell$ servers can jointly compute private key operations whereas no subset of up to $t-1$ servers can similarly compute or otherwise compromise the cryptosystem's security.

Chosen-ciphertext (IND-CCA) security [69,74] is recognized as the *de facto* security notion for public-key encryption. Designing threshold IND-CCA2-secure cryptosystems is non-trivial, and particularly challenging when we aim to combine all desirable properties. In this paper, we are interested in CCA2-secure threshold public-key encryption schemes that are simultaneously: secure under adaptive corruptions (namely, where adversaries can choose whom to corrupt based on the previously obtained information during the protocol), and non-interactive. By "non-interactive" we mean that decryption servers do not communicate with one another in a time consuming protocol, but rather only send a single message to a combiner which gathers these partial decryptions to produce the cleartext. In addition, our goal is to prove security in the standard model (i.e., without the random oracle idealization) and without assuming reliable erasures on behalf of decryption servers. Finally, we also wish to achieve robustness and prevent corrupted servers from hindering the decryption process.

We re-emphasize that we aim at simple non-interactive client/servers protocols where, in order to decrypt a message, a client sends a ciphertext to a decryption server that responds with a decryption share (along with a non-interactive proof of share correctness) without having to talk to other servers. As advocated in [78], such non-interactive protocols are attractive as they require no synchronization among servers, and do not rely on network latency guarantees.

To our knowledge, all solutions that combine all the aforementioned properties [62,65] rely on bilinear maps. In this paper, we consider the problem of schemes realizing the above under other well-established and non-pairing-related standard assumptions.

Non-interactive Schemes. When we aim to avoid interaction during the decryption process in the design of threshold CCA2 schemes, the common stumbling block is that decryption servers often need to know whether an incoming ciphertext is valid or not before releasing their partial decryption result. The early solutions to this problem involved non-interactive zero-knowledge (NIZK) proofs [46,77] of ciphertext well-formedness in the random oracle model. In the standard model, Canetti and Goldwasser [23] thresholdized the Cramer-Shoup cryptosystem [31] by means of a randomized decryption protocol. Their approach involves shared randomizers in order to prevent partial decryptions on invalid ciphertexts from leaking information on the secret key shares. To remove interaction from the process, shareholders have to store a large number of pre-shared randomizers, which entails a prohibitively large storage cost. Cramer, Damgård and Ishai suggested [28] a non-interactive distributed randomization technique but it only supports a small number of servers. Boneh *et al.* [16] observed that, at least for static adversaries, these limitations can be avoided if shared randomizers are generated using non-interactive distributed pseudorandom functions.

In the static corruption setting, generic or partially generic CCA2-secure threshold constructions were proposed in [14,15,42,81]. Boneh, Boyen and Halevi [14] notably came up with the first fully non-interactive realization in the standard model. Their scheme crucially relies on pairings to publicly check the validity of ciphertexts, which drastically simplifies the problem of proving security in the threshold setting. Bilinear maps also provide robustness essentially for free, by making the validity of decryption shares publicly verifiable. Similar applications of the Canetti-Halevi-Katz [24] methodology to threshold cryptography were considered in [18,58]. Wee [81] subsequently laid out a framework for the design of non-interactive threshold signatures and CCA2-secure cryptosystems in the random oracle model under static corruptions.

More recently, Boneh *et al.* [15] introduced a tool, called *universal thresholdizer*, that essentially turns any non-interactive cryptographic scheme (such as public-key encryption, digital signatures or pseudorandom functions) into a threshold variant of the same primitive. Their compiler builds on fully homomorphic encryption (FHE) [50] and notably implies CCA2-secure non-interactive threshold encryption schemes in the static corruption setting.

Adaptive Corruptions. Most threshold cryptosystems (e.g., [14,23,42,46,77]) have been analyzed in a static corruption model, where the adversary commits to the set of corrupted servers *before* the protocol execution. Unfortunately, security under static corruptions does not imply security against more realistic adversaries that can adaptively corrupt servers based on previously and dynamically collected information. Canetti *et al.* [22] put forth adaptively secure key generation protocols for the distributed generation of discrete-log-based keys as well as adaptively secure threshold DSA signatures. Frankel, MacKenzie and Yung [47,48] independently showed different methods to achieve adaptive security. Their techniques were extended [5] to obtain proactive [70] RSA signatures.

The constructions of [22,47,48] inherently require interaction as they rely on the so-called "single inconsistent player" (SIP) technique. The latter consists of transforming a t-out-of-ℓ secret sharing into an additive t-out-of-t sharing of the same secret. In the latter case, only one server (which is randomly chosen ahead of time by the simulator among the ℓ severs) has an inconsistent internal state that causes the simulation to fail if it gets corrupted. Since this occurs with probability $\approx 1/2$, the stuck simulator can rewind the adversary and use different random coins with the hope of avoiding a new corruption of the inconsistent player. The threshold schemes of [47,48] thus proceed by switching from a (t, ℓ) polynomial secret sharing to a (t, t) additive secret sharing by first choosing a set of t participants. If a single participant fails to provide a valid contribution, the whole protocol must restart from scratch.[1] Jarecki and Lysyanskaya [55] extended the SIP technique to eliminate the need for erasures and described an adaptively secure variant of the Canetti-Goldwasser scheme [23]. Abe and Fehr

[1] An alternative approach, suggested in [5,73], requires each participant to store backup shares of other participant's shares in such a way that the missing contributions of faulty servers can be reconstructed. However, it still requires additional interaction.

[2] removed zero-knowledge proofs from the Jarecki-Lysyanskaya construction and proved it secure in (a variant of) the universal composability framework. However, [2,55] both require interaction during the decryption protocol. As in previous threshold variants of Cramer-Shoup, it requires either a large amount of synchronized interaction or a storage of a large number (i.e., proportional to the total number of decryption queries over the lifetime of the system) of pre-shared secrets. As argued in [78], none of the schemes in [1,2,23,55] is a simple, non-interactive, client/server protocol.

An adaptively secure extension of the Boneh-Boyen-Halevi construction [14] was proposed by Libert and Yung [64] using bilinear maps in composite order groups. It was subsequently shown [62,65] that pairing-based NIWI/NIZK arguments [53,56] can be used to remove interaction from threshold variants of Cramer-Shoup while proving security under adaptive corruptions in the standard model. A natural question to ask (from an algebraic perspective aiming not to put all one's eggs in the same [pairing] basket) is whether similarly simple non-interactive adaptively secure systems can be realized in the standard model outside the world of pairing-based cryptography.

Our Contribution. In this paper, we provide IND-CCA-secure non-interactive threshold cryptosystems proven secure in the sense of a game-based definition of *adaptive security* under the Decision Composite Residuosity assumption [71] (DCR) and the Learning-With-Errors (LWE) assumption [75].

Our first construction relies on both assumptions and features ciphertexts that are about as short as in standard (i.e., non-threshold) DCR-based CCA2-secure encryption schemes based on the Cramer-Shoup paradigm [20,32]. Indeed, ciphertexts are only roughly 3 times as large as those of [20]. Our scheme offers at least two advantages over the DCR-based system obtained by applying universal thresholdizers [15] to, e.g., the Camenisch-Shoup cryptosystem [20, Section 3.2]. First, in line with our goal, we can prove adaptive security under a polynomial reduction for $t, \ell = \mathsf{poly}(\lambda)$ without relying on complexity leveraging.[2] Indeed, universal thresholdizers are not known to enable adaptive security under our definition. Second, the scheme implies a more efficient *voting-friendly* threshold cryptosystem [12] in the sense that its ciphertexts can be publicly "downgraded" (by discarding the ciphertext components that ensure CCA2 security) into an additively homomorphic encryption scheme which is much more efficient than the voting-friendly scheme derived from [15,20]. The reason is that the shared decryption algorithm of [15] would proceed by homomorphically evaluating the decryption circuit of the standard Paillier-based scheme [20] over FHE-encrypted Paillier secret keys.

Our second construction relies on the sole LWE assumption. To our knowledge, it is the first adaptively secure non-interactive threshold cryptosystem with CCA2 security in the standard model under a quantum-safe assumption. One caveat is that, analogously to previous LWE-based threshold cryptosystems, it

[2] When $t, \ell = O(\log \lambda)$, statically secure schemes can be proven adaptively secure by guessing the set of corrupted servers upfront.

relies on an LWE assumption with super-polynomial approximation factor. The reason is that, as in all earlier threshold LWE-based constructions [10,15], decryption servers need to add a noise flooding term (which requires a super-polynomial modulus-to-noise ratio) in order to not leak their secret key shares when computing partial decryptions. It remains an open problem to prove adaptive security under a more common LWE assumption with polynomial approximation factor.

Technical Overview. Our schemes build on hash proof systems [32] and can be seen as pairing-free adaptation of constructions proposed by Libert and Yung [65]. In [65], they exploit the property that, in the security proofs of encryption schemes built upon hash proof systems, the simulator always knows the secret keys, which makes it easier to answer adaptive corruption queries (a similar observation was made by Dodis and Fazio [41] in the context of trace-and-revoke schemes). In the threshold setting, the reduction knows all secret key shares and can always provide a consistent internal state for adaptively corrupted servers.

To address the difficulty that valid ciphertexts are not publicly recognizable, [62,65] replaced the designated-verifier NIZK proofs of ciphertext validity [31,32] by publicly verifiable pairing-based NIZK arguments. This eliminates the need for randomized decryption – which was the culprit of interaction in [23,55] – since the shared decryption oracle can just reject invalid ciphertexts. This, in turn, preserves the entropy of the centralized secret key (which is used to create the challenge ciphertext in [31,32]) as decryption queries on valid ciphertexts do not decrease the entropy of secret keys conditionally on the adversary's view. In the challenge ciphertext, the reduction must be able to simulate a fake argument of ciphertext validity while making sure that the adversary cannot come up with such a fake argument in decryption queries. For this purpose, the underlying NIZK argument has to provide one-time simulation-soundness [76].

Our first scheme is a threshold version of (a variant of) an Elgamal-Paillier combination proposed in [20]. The public key contains $h = g^{4N \cdot x} \bmod N^2$, where N is a safe-prime product and $x \in \mathbb{Z}$ is the secret key. Messages $\mathsf{Msg} \in \mathbb{Z}_N$ are encrypted as $(C_0, C_1) = (g^{2N \cdot r}, (1+N)^{\mathsf{Msg}} \cdot h^r) \in (\mathbb{Z}^*_{N^2})^2$ and can be decrypted using x. The security proof of [20] involves a hybrid game where C_0 is sampled as a random quadratic residue (instead of a $2N$-th residue) in $\mathbb{Z}^*_{N^2}$ before computing $C_1 = (1+N)^{\mathsf{Msg}} \cdot C_0^{2x} \bmod N^2$. In order to exploit the entropy of x mod N in the challenge phase, each ciphertext (C_0, C_1) should come with a simulation-sound NIZK proof/argument that C_0 is an N-th residue in $\mathbb{Z}^*_{N^2}$.

This NIZK component can be realized from recent results [21,25,72] on the standard-model instantiability of the Fiat-Shamir paradigm [45]. In our setting, we can use an argument of composite residuosity described by Libert *et al.* [61], which argues soundness in one shot (i.e., without parallel repetitions). However, the latter construction is somewhat an overkill for our purposes as it provides *unbounded* simulation-soundness (USS) while we only need *one-time* simulation-soundness in the context of threshold CCA2 security. We, thus, construct an optimized version of the NIZK argument of [61], where the common reference string (CRS) only contains $O(1)$ Paillier ciphertexts, instead of $O(\lambda)$ in [61].

This new optimized NIZK argument suffices for all applications that only need one-time simulation soundness.[3]

Like its unbounded counterpart [61], our one-time simulation-sound argument adapts a compiler from [60], which builds USS arguments from trapdoor Σ-protocols. In short, these are Σ-protocols in the CRS model where an efficiently computable function $\mathsf{BadChallenge}$ uses a trapdoor to compute the only challenge Chall admitting a valid response z for a given false statement $x \notin \mathcal{L}$ and a given first prover message a. The USS argument of [60] uses a technique due to Damgård [33] that consists of having the prover first send an equivocable commitment to its first Σ-protocol message before opening the commitment in the response z. In [60], the equivocable commitment was replaced by a strengthened version of the $\mathcal{R}$-lossy encryption primitive of Boyle *et al.* [19]. In a nutshell, an $\mathcal{R}$-lossy PKE scheme is a tag-based encryption scheme where ciphertexts are injective for all tags t satisfying some relation $R(K, t)$ (where K is an initialization value chosen at key generation time) and equivocable when $R(K, t) = 0$. By equivocating the ciphertext in all simulated proofs while keeping it extractable in the adversary's fake proof, we can use the extraction trapdoor to compute the $\mathsf{BadChallenge}$ function (in order to ensure soundness via the techniques of [25]), even after having simulated proofs by means of ciphertext equivocation. This can be seen as applying the simulation-sound zero-knowledge techniques of Garay *et al.* [49] in the context of the $\mathsf{BadChallenge}$ function methodology [25].

In [60], it was shown that the underlying equivocable $\mathcal{R}$-lossy PKE scheme can be instantiated from the DCR assumption using public keys comprised of $O(\lambda)$ Paillier ciphertexts. In Sect. 3, we show that, if we only need to simulate *one* argument of a false statement, we can use a more efficient $\mathcal{R}$-lossy PKE scheme for a different relation allowing for constant-size public keys. While [61] uses the bit-matching relation of [19] which incurs long public keys as it must be combined with admissible hash functions [13], we can simply use the inequality relation where $R(K, t) = 1$ if and only if $K \neq t$. In our DCR-based instantiation, we can thus encrypt/commit to $\mathsf{Msg} \in \mathbb{Z}_N$ under the tag t by computing $\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}} \cdot r^N \bmod N^2$, which can be decrypted as a standard Paillier ciphertext when N divides the order of $u^t \cdot v$. When $u^t \cdot v$ is an N-th residue, we can equivocate ct by finding $r \in \mathbb{Z}_N^*$ that explains ct as an encryption of an arbitrary plaintext. By suitably programming $u, v \in \mathbb{Z}_{N^2}^*$, we can make sure that $u^t \cdot v$ is an N-th residue for one specific tag $t = K$. Importantly, we need to equivocate without knowing the factorization of N since, in our application to simulation-soundness, we rely on the DCR assumption to switch between settings where either only one tag is equivocable or all tags are equivocable.

We note that the above tools do not quite make valid ciphertexts publicly recognizable because the NIZK argument only guarantees that C_0 is a composite residue without proving that it is also a square in $\mathbb{Z}_{N^2}^*$. However, it does not affect the application to CCA2 security since decryption servers can simply square C_0

[3] Faust *et al.* [43] showed that Fiat-Shamir provides simulation-soundness "for free" in the ROM. However, their proof crucially relies on the random oracle modeling of hash functions and it is not known to immediately carry over to the standard model.

themselves to make sure that C_0^2 lives in the subgroup of $2N$-th residues before releasing decryption shares $C_0^{2\cdot \mathsf{sk}_i}$.

Our LWE-based construction relies on the dual Regev cryptosystem [51], where public keys contain random matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{U} = \mathbf{A}\cdot\mathbf{R} \in \mathbb{Z}_q^{n\times L}$, for some $n, m, L \in \mathsf{poly}(\lambda)$ such that $n < m$, and secret keys are small-norm integer matrices $\mathbf{R} \in \mathbb{Z}^{m\times L}$. Since the columns of $\mathbf{R}$ have a lot of entropy conditionally on $(\mathbf{A}, \mathbf{U})$, it is tempting to adapt the approach of our DCR-based system and use LWE in an hash-proof-like fashion (as previously done in, e.g., [6]). However, this requires preventing the adversary from inferring information on $\mathbf{R}$ by making decryption queries on ill-formed ciphertexts. This cannot be achieved via designated-verifier NIZK proofs [32] since known LWE-based hash proof systems (e.g., [57,82]) do not provide smoothness in the worst-case. Namely, nothing is guaranteed on the unpredictability of $\mathbf{R}^\top\mathbf{c}_0$ when $\mathbf{c}_0$ is neither a vector of LWE samples for $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ nor a uniform vector over $\mathbb{Z}_q^m$, but something in between (e.g., a vector $\mathbf{c}_0 = \mathbf{A}^\top\cdot\mathbf{s} + \mathbf{e}_0$ where $\mathbf{e}_0 \in \mathbb{Z}^m$ is slightly too large).

To address the problem of showing that $\mathbf{c}_0$ is well-formed (i.e., of the form $\mathbf{c}_0 = \mathbf{A}^\top\cdot\mathbf{s}+\mathbf{e}_0$ for a small enough $\mathbf{e}_0 \in \mathbb{Z}^m$), we replace the designated-verifier NIZK proof by a Fiat-Shamir-based [45] publicly verifiable NIZK argument, which is known to provide soundness in the standard model under the LWE assumption [72]. To avoid relying on a generic Karp reduction to the Graph Hamiltonicity language used in [25], we rely on the simulation-sound NIZK argument of Libert *et al.* [60, Appendix G] which allows showing that a vector $\mathbf{c}_0$ is indeed of the form $\mathbf{c}_0 = \mathbf{A}^\top\cdot\mathbf{s}+\mathbf{e}_0$ for a small $\mathbf{e}_0 \in \mathbb{Z}^m$. Since their construction provides publicly verifiable arguments, its soundness property does not rely on the entropy of a verifier's secret key and bypasses the difficulties arising from the use of designated-verifier NIZK proofs. In particular, it keeps the verifier from accepting proofs for vectors $\mathbf{c}_0 = \mathbf{A}^\top\cdot\mathbf{s}+\mathbf{e}_0$ where $\mathbf{e}_0$ is only slightly too large, which preserves the entropy of the centralized secret key $\mathbf{R} \in \mathbb{Z}^{m\times\ell}$.

In the threshold setting, both schemes share their secret keys using the linear integer secret sharing (LISS) primitive of Damgård and Thorbek [35], which are similar to linear secret sharing schemes except that they work over $\mathbb{Z}$. In our LWE-based construction, we crucially exploit the fact that LISS schemes have small linear reconstruction coefficients that can multiply decryption shares without blowing up the underlying noise terms. We could have alternatively used $\{0,1\}$-linear secret sharing (which can also express monotone Boolean formulas [59]) as in [15]. However, as observed in [63] in the adaptive corruption setting, LISS nicely interact with discrete Gaussian distributions and make it easier to analyze the remaining entropy of shared secret keys after all decryption queries and corruption queries. Indeed, our DCR-based TPKE bears similarities to the inner product functional encryption scheme of Agrawal *et al.* [4] in that it samples secret keys $x \in \mathbb{Z}$ from a Gaussian distribution over the integers. By sharing them with a LISS, we can adapt arguments used in [4,63] in order to assess the entropy of secret keys after all queries.

Related Work. Back in 2001, Fouque and Pointcheval [46] used the Naor-Yung paradigm [69] to construct a CCA2-secure threshold cryptosystem under the DCR assumption in the random oracle model. In the full version of the paper, we show, as a comment, that the proof of IND-CCA security of [46] is actually incorrect as an adversary can break the soundness of the proof of plaintext equalities between Paillier ciphertexts with different moduli. It can be fixed by having the encryptor prove that the plaintext is a positive integer smaller than both moduli.

The first LWE-based threshold encryption scheme was proposed by Bendlin and Damgård [10] who showed a threshold version of Regev's cryptosystem [75]. Xie *et al.* [83] gave a threshold CCA-secure realization where the size of public keys and ciphertexts grows at least linearly with the number of servers. Boneh *et al.* gave a compiler [15] that turns any IND-CCA secure into a non-interactive threshold variant thereof using fully homomorphic encryption. Bendlin *et al.* [11] considered lattice-based threshold signatures and IBE schemes. However, the servers can only compute an a priori bounded number of non-interactive private key operations without using interaction. Libert *et al.* [63] described non-interactive threshold pseudorandom functions from LWE. Our LWE-based TPKE and its security proof are actually inspired by their use of LISS schemes.

Organization. In Sect. 2, we first recall some definitions and tools that will be used in our constructions. Section 3 then presents our one-time simulation-sound NIZK arguments, which builds on our DCR-based $\mathcal{R}$-lossy PKE scheme described in Sect. 3.1. Our DCR-based threshold cryptosystem is explained in Sect. 4. Its variant based on the sole LWE assumption is given in the full version of the paper. For simplicity, we first present non-robust variants of both schemes. In the full version of the paper [40], we show that standard techniques can be applied to achieve robustness against malicious adversaries.

2 Background and Definitions

2.1 Lattices

For any $q \geq 2$, $\mathbb{Z}_q$ denotes the ring of integers with addition and multiplication modulo q. If $\mathbf{x} \in \mathbb{R}^n$ is a vector, $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} x_i^2}$ denotes its Euclidean norm and $\|\mathbf{x}\|_\infty = \max_i |x_i|$ its infinity norm. If $\mathbf{M}$ is a matrix over $\mathbb{R}$, then $\|\mathbf{M}\| := \sup_{\mathbf{x}\neq 0} \frac{\|\mathbf{M}\mathbf{x}\|}{\|\mathbf{x}\|}$ and $\|\mathbf{M}\|_\infty := \sup_{\mathbf{x}\neq 0} \frac{\|\mathbf{M}\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty}$ denote its induced norms. For a finite set S, $U(S)$ stands for the uniform distribution over S. If X and Y are distributions over the same domain, $\Delta(X, Y)$ denotes their statistical distance.

Let $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ be a symmetric positive-definite matrix, and $\mathbf{c} \in \mathbb{R}^n$. We define the Gaussian function on $\mathbb{R}^n$ by $\rho_{\mathbf{\Sigma},\mathbf{c}}(\mathbf{x}) = \exp(-\pi(\mathbf{x}-\mathbf{c})^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{c}))$ and if $\mathbf{\Sigma} = \sigma^2 \cdot \mathbf{I}_n$ and $\mathbf{c} = \mathbf{0}$ we denote it by ρ_σ. For an n dimensional lattice $\Lambda \subset \mathbb{R}^n$ and for any lattice vector $\mathbf{x} \in \Lambda$ the discrete Gaussian is defined by $\rho_{\Lambda,\mathbf{\Sigma},\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\mathbf{\Sigma},\mathbf{c}}}{\rho_{\mathbf{\Sigma},\mathbf{c}}(\Lambda)}$.

For an n-dimensional lattice Λ, we define $\eta_\varepsilon(\Lambda)$ as the smallest $r > 0$ such that $\rho_{1/r}(\widehat{\Lambda} \setminus \mathbf{0}) \leq \varepsilon$ with $\widehat{\Lambda}$ denoting the dual of Λ, for any $\varepsilon \in (0,1)$.

For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we define $\Lambda^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q\}$ and $\Lambda(\mathbf{A}) = \mathbf{A}^\top \cdot \mathbb{Z}^n + q\mathbb{Z}^m$. For an arbitrary vector $\mathbf{u} \in \mathbb{Z}_q^n$, we also define the shifted lattice $\Lambda^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q\}$.

We now recall the definition of the Learning-With-Errors (LWE) assumption introduced by Regev [75].

Definition 2.1 (LWE assumption). *Let $m \geq n \geq 1$, $q \geq 2$ and $\alpha \in (0,1)$ be functions of a security parameter λ. The LWE problem consists in distinguishing between the distributions $(\mathbf{A}, \mathbf{A}\mathbf{s}+\mathbf{e})$ and $U(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$, where $\mathbf{A} \sim U(\mathbb{Z}_q^{m \times n})$, $\mathbf{s} \sim U(\mathbb{Z}_q^n)$ and $\mathbf{e} \sim D_{\mathbb{Z}^m, \alpha q}$.*

Lemma 2.2 ([51, Theorem 4.1]). *There is a PPT algorithm that, given a basis $\mathbf{B}$ of an n-dimensional $\Lambda = \Lambda(\mathbf{B})$, a parameter $s > \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log n})$, and a center $\mathbf{c} \in \mathbb{R}^n$, outputs a sample from a distribution statistically close to $D_{\Lambda, s, \mathbf{c}}$.*

Lemma 2.3 ([68], Lemma 4.4). *For $\sigma = \omega(\sqrt{\log n})$, there exists a negligible function $\epsilon = \epsilon(n)$ such that $\Pr_{\mathbf{x} \hookleftarrow D_{\mathbb{Z}^n, \sigma}}[\|\mathbf{x}\| > \sigma\sqrt{n}] \leq \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-n}$.*

Lemma 2.4 ([63, Lemma 2.6]). *Let $\epsilon \in (0,1), c \in \mathbb{R}$ and $\sigma > 0$, such that $\sigma \geq \sqrt{\ln 2(1+1/\epsilon)/\pi}$. Then $H_\infty(D_{\mathbb{Z}, \sigma, c}) \geq \log \sigma - \log\left(1 + \frac{2\epsilon}{1-\epsilon}\right)$. For $\sigma = \Omega(\sqrt{n})$, we get $H_\infty(D_{\mathbb{Z}, \sigma, c}) \geq \log(\sigma) - 2^{-n}$.*

Lemma 2.5 ([44]). *Let $\beta > 0, q \in \mathbb{Z}$ and $y \in \mathbb{Z}$. Then, the following holds: $\Delta(D_{\mathbb{Z}_q, \beta \cdot q, 0}, D_{\mathbb{Z}_q, \beta \cdot q, y}) \leq \frac{|y|}{\beta q}$.*

Lemma 2.6 ([67, Theorem 2]). *There exists an efficient randomized algorithm $\mathsf{TrapGen}(1^n, 1^m, q)$ that given any integers $n \geq 1, q \geq 2$ and sufficiently large $m = O(n \log q)$ outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a trapdoor $\mathbf{T}_\mathbf{A}$ such that the distribution of $\mathbf{A}$ is statistically close to uniform.*

Lemma 2.7 (Adapted from [51, corollary 2.8]). *Let $\Lambda' \subseteq \Lambda \subseteq \mathbb{R}^n$ be two lattices with the same dimension. Let $\epsilon \in (0, 1/2)$. Then, for any $c \in \mathbb{R}^n$ and any $\sigma \geq \eta_\epsilon(\Lambda')$, the distribution $D_{\Lambda, \sigma, c} \bmod \Lambda'$ is within statistical distance 2ϵ from the uniform distribution over Λ/Λ'.*

2.2 Composite Residuosity Assumption

We now recall Paillier's Composite Residuosity assumption and its variant considered by Damgård and Jurik.

Definition 2.8 ([34,71]). *Let integers $N = pq$ and $s > 1$ for primes p, q. The s-**Decision Composite Residuosity** (s-DCR) assumption states that the distributions $\{x = w^{N^s} \bmod N^{s+1} \mid w \leftarrow U(\mathbb{Z}_N^\star)\}$ and $\{x \mid x \leftarrow U(\mathbb{Z}_{N^{s+1}}^\star)\}$ are computationally indistinguishable.*

It is known [34] that the s-DCR assumption is equivalent to the standard 1-DCR of [71] for any $s > 1$.

2.3 Linear Integer Secret Sharing

This section recalls the concept of linear integer secret sharing (LISS), as defined by Damgård and Thorbek [35]. Definitions below are taken from [79] where the secret to be shared lives in an interval $[-2^l, 2^l]$ centered in 0, for some $l \in \mathbb{N}$.

Definition 2.9. *A **monotone** access structure on $[\ell]$ is a non-empty collection $\mathbb{A}$ of sets $A \subseteq [\ell]$ such that $\emptyset \notin \mathbb{A}$ and, for all $A \in \mathbb{A}$ and all sets B such that $A \subseteq B \subseteq [\ell]$, we have $B \in \mathbb{A}$. For an integer $t \in [\ell]$, the **threshold**-t access structure $T_{t,\ell}$ is the collection of sets $A \subseteq [\ell]$ such that $|A| \geq t$. Sets $A \in \mathbb{A}$ are called qualified and sets $B \notin \mathbb{A}$ are called forbidden.*

Let $P = [\ell]$ be a set of shareholders. In a LISS scheme, a dealer D wants to share a secret s in a publicly known interval $[-2^l, 2^l]$. To this end, D uses a share generating matrix $\mathbf{M} \in \mathbb{Z}^{d \times e}$ and a random vector $\boldsymbol{\rho} = (s, \rho_2, \ldots, \rho_e)^\top$, where s is the secret to be shared and $\{\rho_i\}_{i=2}^e$ are randomly sampled in $[-2^{l_0+\lambda}, 2^{l_0+\lambda}]^{e-1}$, for some $l_0 \geq l \in \ell$. Usually, the distribution of the ρ_i is uniform but, in the following, we will set $l_0 = l$ and $\rho_i \hookleftarrow D_{\mathbb{Z},\sigma}$. The dealer D computes a vector $\boldsymbol{s} = (s_1, \ldots, s_d)^\top$ of share units as $\boldsymbol{s} = (s_1, \ldots, s_d)^\top = \mathbf{M} \cdot \boldsymbol{\rho} \in \mathbb{Z}^d$. Each party in $P = \{1, \ldots, \ell\}$ is assigned a set of share units. Letting $\psi : \{1, \ldots, d\} \to P$ be a surjective function, the i-th share unit s_i is assigned to the shareholder $\psi(i) \in P$, in which case player $\psi(i)$ is said to own the i-th row of $\mathbf{M}$. If $A \subseteq P$ is a set of shareholders, $\mathbf{M}_A \in \mathbb{Z}^{d_A \times e}$ denotes the set of rows jointly owned by A. Likewise, $\boldsymbol{s}_A \in \mathbb{Z}^{d_A}$ denotes the restriction of $\boldsymbol{s} \in \mathbb{Z}^d$ to the coordinates jointly owned by the parties in A. The j-th shareholder's share consists of $\boldsymbol{s}_{\psi^{-1}(j)} \in \mathbb{Z}^{d_j}$, so that it receives $d_j = |\psi^{-1}(j)|$ out of the $d = \sum_{j=1}^{\ell} d_j$ share units. The *expansion rate* $\mu = d/\ell$ is defined to be the average number of share units per player.

There exist security notions for LISS schemes but, since we do not explicitly rely on them, we omit their exposition for conciseness.

To construct LISS schemes, Damgård and Thorbek [35] used integer span programs [30].

Definition 2.10 ([30]). *An integer span program (ISP) is a tuple formed by three elements $\mathcal{M} = (\mathbf{M}, \psi, \boldsymbol{\varepsilon})$, where $\mathbf{M} \in \mathbb{Z}^{d \times e}$ is an integer matrix whose rows are labeled by a surjective function $\psi : \{1, \ldots, d\} \to \{1, \ldots, \ell\}$ and $\boldsymbol{\varepsilon} = (1, 0, \ldots, 0)$ is called target vector. The size of $\mathcal{M}$ is the number of rows d in $\mathbf{M}$.*

Definition 2.11. *Let Γ be a monotone access structure and let $\mathcal{M} = (\mathbf{M}, \psi, \boldsymbol{\varepsilon})$ an integer span program. Then, $\mathcal{M}$ is an ISP for Γ if it computes Γ: namely, for all $A \subseteq \{1, \ldots, \ell\}$, the following conditions hold:*

1. *If $A \in \Gamma$, there is a reconstruction vector $\boldsymbol{\lambda} \in \mathbb{Z}^{d_A}$ such that $\boldsymbol{\lambda}^\top \cdot \mathbf{M}_A = \boldsymbol{\varepsilon}^\top$.*
2. *If $A \notin \Gamma$, there exists $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_e)^\top \in \mathbb{Z}^e$ such that $\mathbf{M}_A \cdot \boldsymbol{\kappa} = \mathbf{0} \in \mathbb{Z}^d$ and $\boldsymbol{\kappa}^\top \cdot \boldsymbol{\varepsilon} = 1$. In this case, $\boldsymbol{\kappa}$ is called a sweeping vector for A.*

We also define $\kappa_{\max} = \max\{|a| \mid a \text{ is an entry in some sweeping vector}\}$.

Damgård and Thorbek [35] observed that a LISS can be built by setting the share generating matrix to be the matrix $\mathbf{M}$ of an ISP $\mathcal{M} = (\mathbf{M}, \psi, \varepsilon)$ that computes the access structure Γ. We may then specify a LISS scheme $\mathcal{L} = (\mathcal{M} = (\mathbf{M}, \psi, \varepsilon), \Gamma, \mathcal{R}, \mathcal{K})$ by an ISP for the access structure Γ, a space $\mathcal{R}$ of reconstruction vectors satisfying Condition 1 of Definition 2.11, and a space $\mathcal{K}$ of sweeping vectors satisfying Condition 2.

The last step is building an ISP for any access structure with small reconstruction vectors and small sweeping vectors. Damgård and Thorbek showed in [35] that LISS schemes can be obtained from [9,30]. While the Benaloh-Leichter (BL) secret sharing [9] was designed to work over finite groups, it was generalized in [35] to share integers using access structures consisting of monotone Boolean formulas. In turn, this implies a LISS scheme for any threshold access structure by applying a result of Valiant [52,80]. Their LISS scheme built upon Benaloh-Leichter [9] satisfies what we want: as can be observed from [35, Lemma 4], every coefficient of any reconstruction vector $\boldsymbol{\lambda}$ lives in $\{-1, 0, 1\}$ and [35, Lemma 5] shows that $\kappa_{\max} = 1$. Let a monotone Boolean formula f, then the BL-based technique allows us to build binary share distribution matrices $\mathbf{M} \in \{0,1\}^{d \times e}$ such that $d, e = O(\text{size}(f))$. Moreover they have at most $\text{depth}(f) + 1$ non-zero entries, so that each share unit s_i has magnitude $O(2^{l_0+\lambda} \cdot \text{depth}(f))$.

Finally, Valiant's result [80] implies the existence of a monotone Boolean formula of the threshold-t function $T_{t,\ell}$, which has size $d = O(\ell^{5.3})$ and depth $O(\log \ell)$. Recall that each player will receive about d/ℓ rows of $\mathbf{M}$ on average, then the average share size is $O(\ell^{4.3} \cdot (l_0 + \lambda + \log\log \ell))$ bits. Valiant's construction was improved by Hoory *et al.* [54] who gave a monotone formula of size $O(\ell^{1+\sqrt{2}})$ and depth $O(\log \ell)$ for the majority function.[4] This in turn reduces the average share size to $O(\ell^{\sqrt{2}} \cdot (l_0 + \lambda + \log\log \ell))$ bits.

2.4 Threshold PKE

In this section, we recall the TPKE syntax defined by Boneh *et al.* [15].

Definition 2.12 (Threshold PKE). *Let P be a party of ℓ servers and $\mathbb{S}$ be a class of efficient monotone access structure on P. A Threshold PKE scheme (TPKE) for some message space $\mathcal{M}$ is then a tuple of efficient* PPT *algorithms* (Keygen, Encrypt, PartDec, PartVerify, Combine) *with the following specifications:*

- $\mathsf{Keygen}(1^\lambda, \mathbb{A}) \rightarrow (\mathsf{pp}, \mathsf{ek}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell)$*: On input a security parameter λ, $\mathbb{A} \in \mathbb{S}$ an access structure, the algorithm outputs a set of public parameters* pp *(which are implicit in the inputs of all other algorithms), a public key* pk *and a set of secret key shares* $\mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell$.
- $\mathsf{Encrypt}(\mathsf{pk}, \mathsf{Msg}) \rightarrow \mathsf{ct}$*: On input the public parameters* pp*, the encryption key* ek *and a message* $\mathsf{Msg} \in \mathcal{M}$*, the algorithm outputs a ciphertext* ct.

[4] Note that a threshold-t function can be obtained from the majority function by fixing the desired number of input bits, so that we need a majority function of size $\leq 2\ell$ to construct a threshold function $T_{t,\ell}$.

- $\mathsf{PartDec}(\mathsf{pk}, \mathsf{ct}, \mathsf{sk}_i) \rightarrow \mu_i$*: Given public parameters* pp*, a ciphertext* ct *and a secret key share* sk_i*, this algorithm outputs a partial decryption* μ_i*.*
- $\mathsf{PartVerify}(\mathsf{pk}, \mathsf{ct}, \mu_i) \rightarrow \mathsf{b} \in \{0,1\}$*: On input of public parameters* pp*, a ciphertext* ct *and a partial decryption* μ_i*, this algorithm outputs a bit* b*.*
- $\mathsf{Combine}\,(\mathsf{pk}, B = (\mathcal{S}, \{\phi(\mu_i)\}_{i\in\mathcal{S}}), \mathsf{ct}) \rightarrow \mathsf{Msg}'$*: Given public parameters and a set of images of* ϕ *of partial decryptions, the algorithm outputs a message* $\mathsf{Msg}' \in \mathcal{M}$*. The function* ϕ *is public and deterministic.*[5]

The goal is now to construct a TPKE scheme that satisfies the following compactness, correctness and requirements.

Definition 2.13 (Compactness [15]). *A TPKE scheme satisfies compactness if there exist polynomials* P *and* Q *such that* $\forall\lambda, \forall\mathbb{A} \in \mathbb{S}, |\mathsf{pk}| \leq P(\lambda) \wedge |\mathsf{ct}| \leq Q(\lambda)$*, where* $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$ *and the ciphertext is generated with* $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{Msg})$ *for any* $\mathsf{Msg} \in \mathcal{M}$*.*

Definition 2.14 (Decryption Correctness). *A TPKE provides decryption correctness if the following holds. For any* $\lambda \in \mathbb{N}$*, any access structure* $\mathbb{A} \in \mathbb{S}$*, any set* $\mathcal{S} \in \mathbb{A}$ *and any message* $\mathsf{Msg} \in \mathcal{M}$*, if we run* $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$*,* $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{Msg})$ *and then* $\mu_i \leftarrow \mathsf{PartDec}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct}), \forall i \in \mathcal{S}$*, we have* $\Pr[\mathsf{Combine}\,(\mathsf{pk}, (\mathcal{S}, \{\phi(\mu_i)\}_{i\in\mathcal{S}})\,, \mathsf{ct}) = \mathsf{Msg}] = 1 - \mathrm{negl}(\lambda)$*.*

Definition 2.15 (Partial Verification Correctness). *A TPKE provides partial verification correctness if the following holds. For any* $\lambda \in \mathbb{N}$*, any* $\mathbb{A} \in \mathbb{S}$*, any* $\mathcal{S} \in \mathbb{A}$ *and any message* $\mathsf{Msg} \in \mathcal{M}$*, if we run* $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$*,* $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{Msg})$ *and* $\mu_i \leftarrow \mathsf{PartDec}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct}), \forall i \in \mathcal{S}$*, then* $\Pr[\mathsf{PartVerify}(\mathsf{pk}, \mathsf{ct}, \mu_i) = 1] = 1 - \mathrm{negl}(\lambda)$*.*

We can now define chosen-ciphertext security in a model allowing the adversary to adaptively corrupt decryption servers.

Definition 2.16 (Adaptive-CCA security for TPKE). *A TPKE scheme provides chosen-ciphertext security under adaptive corruptions if no PPT adversary* $\mathcal{A}$ *has non-negligible advantage in the following game.*

1. *On input the the security parameter* λ*,* $\mathcal{A}$ *chooses an access structure* $\mathbb{A}$*.*
2. *The challenger generates* $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$*. It sends* $(\mathsf{pp}, \mathsf{pk})$ *to* $\mathcal{A}$ *and initializes an empty set* $\mathcal{C} = \emptyset$*.*
3. $\mathcal{A}$ *can adaptively interleave the following queries:*
   - ***Corruption****:* $\mathcal{A}$ *sends the challenger an index* $i \in [\ell]$*. The challenger replies by returning the share* sk_i *and updating the set* $\mathcal{C} = \mathcal{C} \cup \{i\}$*.*
   - ***Partial Decryption****:* $\mathcal{A}$ *chooses an index* $i \in [\ell]$ *and a ciphertext* ct *and the challenger returns a partial decryption* $\mu_i \leftarrow \mathsf{PartDec}(\mathsf{pk}, \mathsf{sk}_i, \mathsf{ct})$*.*
4. $\mathcal{A}$ *chooses* $\mathsf{Msg}_0^\star, \mathsf{Msg}_1^\star \in \mathcal{M}$*. The challenger replies with a challenge ciphertext* $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{Msg}_b^\star)$*, where* $b \hookleftarrow U(\{0,1\})$ *is a random bit.*

[5] It helps defining robustness. For non-robust TPKE, ϕ is the identity function.

5. *$\mathcal{A}$ makes more corruption and partial decryption queries subject to the following condition which must be satisfied at any time. Let $\mathcal{C} \subset [\ell]$ the set of corrupted servers and let $\mathcal{C}^*$ the subset of indexes $j \in [\ell]$ such that $\mathcal{A}$ made a decryption query of the form (j, ct^*). Then, it is required that $\mathcal{C} \cup \mathcal{C}^* \notin \mathbb{A}$.*
6. *The experiment ends with $\mathcal{A}$ outputting a bit $b' \in \{0,1\}$.*

The advantage of $\mathcal{A}$ is defined as $\mathbf{Adv}^{\text{ind-cca}}(\mathcal{A}) := \left|\Pr[b' = b] - \frac{1}{2}\right|$.

We now recall the notion of *robustness*, which informally captures that no malicious adversary can prevent a honest majority from decrypting a valid ciphertext.

Definition 2.17 ([15]). *A TPKE scheme satisfies* **robustness** *if no PPT adversary $\mathcal{A}$ can cause the following experiment $\mathsf{Expt}^{\text{robust}}_{\mathcal{A},\mathsf{TPKE}}(1^\lambda)$ to output 1 with non-negligible probability.*

1. *On input the security parameter λ, $\mathcal{A}$ chooses an access structure $\mathbb{A}$.*
2. *The challenger samples $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$ and provides $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell)$ to $\mathcal{A}$.*
3. *$\mathcal{A}$ outputs a partial decryption forgery $\left(\mathsf{ct}^\star, \mu_i^\star, i\right)$, where $i \in [\ell]$.*
4. *The experiment outputs 1 if we have $\phi(\hat{\mu}_i^\star) \neq \phi(\mathsf{PartDec}(\mathsf{pk}, \mathsf{sk}_i, \mathsf{ct}^\star))$ while $\mathsf{PartVerify}(\mathsf{pk}, \mathsf{ct}^\star, \mu_i^\star) = 1$.*

We note that the function ϕ allows considering as robust a TPKE such that $\mu_i^\star = (\hat{\mu}_i^\star, \pi_i^\star)$ and where $\mathsf{Combine}$ only runs on $\hat{\mu}_i^\star$ and not on $\pi_i^\star$. While, given $(\hat{\mu}_i^\star, \pi_i^\star)$, $\mathsf{Combine}$ could have simply striped $\pi_i^\star$, such formalization would prevent showing as robust a TPKE where $\hat{\mu}_i^\star$ is a word in an admissible language and $\pi_i^\star$ is a probabilistic membership argument whose validity, moreover, does not necessarily ensure that $\pi_i^\star$ is in the range of honestly computed arguments. Thanks to ϕ, such case will not be artificially discarded.

A weaker robustness notion, a.k.a. consistency [14], captures the robustness of schemes where the word $\hat{\mu}_i^\star$ itself is probabilistic and whose validity tolerates a gap with respect to honestly computed statements. Here, we will focus on the (stronger) notion of robustness.

2.5 Correlation Intractable Hash Functions

We consider unique-output efficiently searchable relations [21].

Definition 2.18. *A relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ is* **searchable** *in time T if there exists a function $f : \mathcal{X} \to \mathcal{Y}$ which is computable in time T and such that, if there exists y such that $(x, y) \in R$, then $f(x) = y$.*

Let $\lambda \in \mathbb{N}$ a security parameter. A hash family with input length $n(\lambda)$ and output length $m(\lambda)$ is a collection $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ of keyed functions induced by efficient algorithms $(\mathsf{Gen}, \mathsf{Hash})$, where $\mathsf{Gen}(1^\lambda)$ outputs a key $k \in \{0,1\}^{s(\lambda)}$ and $\mathsf{Hash}(k, x)$ computes $h_\lambda(k, x) \in \{0,1\}^{m(\lambda)}$.

Definition 2.19. *For a relation ensemble* $\{R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}\}$*, a hash function family* $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ *is* R**-correlation intractable** *if, for any probabilistic polynomial time (PPT) adversary* $\mathbb{A}$*, we have* $\Pr\left[k \leftarrow \mathsf{Gen}(1^\lambda)),\ x \leftarrow \mathcal{A}(k) : (x, h_\lambda(k,x)) \in R\right] = \mathsf{negl}(\lambda)$.

Peikert and Shiehian [72] described a correlation-intractable hash family for any searchable relation (in the sense of Definition 2.18) defined by functions f of bounded depth. When f is computable by a branching program, their construction relies on the standard SIS assumption with polynomial approximation factors. Under the LWE assumption with polynomial approximation factors, their bootstrapping theorem allows handling arbitrary bounded-depth functions.

2.6 Trapdoor Σ-protocols

Canetti *et al.* [25] considered a definition of Σ-protocols that slightly differs from the usual formulation [27,29].

Definition 2.20 (Adapted from [7,25]). *Let a language* $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$ *associated with two NP relations* $R_{\mathsf{zk}}, R_{\mathsf{sound}}$*. A 3-move interactive proof system* $\Pi = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ *in the common reference string model is a Gap* Σ*-protocol for* $\mathcal{L}$ *if it satisfies the following conditions:*

- **3-Move Form:** P *and* V *both take as input* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$*, with* $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ *and* $\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$*, and a statement* x *and proceed as follows: (i)* P *takes in* $w \in R_{\mathsf{zk}}(x)$*, computes* $(\mathbf{a}, st) \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$ *and sends* $\mathbf{a}$ *to the verifier; (ii)* V *sends back a random challenge* Chall *from the challenge space* $\mathcal{C}$*; (iii)* P *finally sends a response* $\mathbf{z} = \mathsf{P}(\mathsf{crs}, x, w, \mathbf{a}, \mathsf{Chall}, st)$ *to* V*; (iv) On input of* $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$*,* V *outputs* 1 *or* 0*.*
- **Completeness:** *If* $(x, w) \in R_{\mathsf{zk}}$ *and* P *honestly computes* $(\mathbf{a}, \mathbf{z})$ *for a challenge* Chall*,* $\mathsf{V}(\mathsf{crs}, x, (\mathbf{a}, \mathsf{Chall}, \mathbf{z}))$ *outputs* 1 *with probability* $1 - \mathsf{negl}(\lambda)$*.*
- **Special zero-knowledge:** *There is a PPT simulator* ZKSim *that inputs* crs*,* $x \in \mathcal{L}_{\mathsf{zk}}$ *and a challenge* $\mathsf{Chall} \in \mathcal{C}$*. It outputs* $(\mathbf{a}, \mathbf{z}) \leftarrow \mathsf{ZKSim}(\mathsf{crs}, x, \mathsf{Chall})$ *such that* $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ *is computationally indistinguishable from a real transcript with challenge* Chall *(for* $w \in R_{zk}(x)$*).*
- **Special soundness:** *For any CRS* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$ *obtained as* $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^\lambda)$*,* $\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$*, any* $x \notin \mathcal{L}_{\mathsf{sound}}$*, and any first message* $\mathbf{a}$ *sent by* P*, there is at most one challenge* $\mathsf{Chall} = f(\mathsf{crs}, x, \mathbf{a})$ *for which an accepting transcript* $(\mathsf{crs}, x, \mathbf{a}, \mathsf{Chall}, \mathbf{z})$ *exists for some third message* $\mathbf{z}$*. The function* f *is called the "bad challenge function" of* Π*. That is, if* $x \notin \mathcal{L}_{\mathsf{sound}}$ *and the challenge differs from the bad challenge, the verifier never accepts.*

Definition 2.20 is taken from [25] and relaxes the standard special soundness property in that extractability is not required. Instead, it considers a bad challenge function f, which may not be efficiently computable. Canetti *et al.* [25] define *trapdoor* Σ-protocols as Σ-protocols where the bad challenge function is efficiently computable using a trapdoor. Here, we use a definition where the CRS and the trapdoor may depend on the language.

The common reference string $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$ consists of a fixed part par and a language-dependent part $\mathsf{crs}_{\mathcal{L}}$ which is generated as a function of par and a language parameter $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$.

Definition 2.21 (Adapted from [25]). *A Σ-protocol $\Pi = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ with bad challenge function f for a trapdoor language $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$ is a* **trapdoor Σ-protocol** *if it satisfies the properties of Definition 2.20 and there exist PPT algorithms $(\mathsf{TrapGen}, \mathsf{BadChallenge})$ with the following properties.*

- $\mathsf{Gen}_{\mathsf{par}}$ *inputs $\lambda \in \mathbb{N}$ and outputs public parameters $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^{\lambda})$.*
- $\mathsf{Gen}_{\mathcal{L}}$ *is a randomized algorithm that, on input of public parameters par, outputs the language-dependent part $\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$ of $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$.*
- $\mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$ *takes as input public parameters par and a membership-testing trapdoor $\tau_{\mathcal{L}}$ for the language $\mathcal{L}_{\mathsf{sound}}$. It outputs a common reference string $\mathsf{crs}_{\mathcal{L}}$ and a trapdoor $\tau_{\Sigma} \in \{0,1\}^{\ell_{\tau}}$, for some $\ell_{\tau}(\lambda)$.*
- $\mathsf{BadChallenge}(\tau_{\Sigma}, \mathsf{crs}, x, \mathbf{a})$ *takes in a trapdoor τ_{Σ}, a CRS $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$, an instance x, and a first prover message $\mathbf{a}$. It outputs a challenge Chall.*

In addition, the following properties are required.

- **CRS indistinguishability:** *For any $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^{\lambda})$, and any trapdoor $\tau_{\mathcal{L}}$ for the language $\mathcal{L}$, an honestly generated $\mathsf{crs}_{\mathcal{L}}$ is computationally indistinguishable from a CRS produced by $\mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$. Namely, for any aux and any PPT distinguisher $\mathcal{A}$, we have*

$$\begin{aligned}\mathbf{Adv}^{\text{indist-}\Sigma}_{\mathcal{A}}(\lambda) := |\Pr[\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}) : \mathcal{A}(\mathsf{par}, \mathsf{crs}_{\mathcal{L}}) = 1] \\ - \Pr[(\mathsf{crs}_{\mathcal{L}}, \tau_{\Sigma}) \leftarrow \mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}}) : \mathcal{A}(\mathsf{par}, \mathsf{crs}_{\mathcal{L}}) = 1]| \leq \mathsf{negl}(\lambda).\end{aligned}$$

- **Correctness:** *There exists a language-specific trapdoor $\tau_{\mathcal{L}}$ such that, for any instance $x \notin \mathcal{L}_{\mathsf{sound}}$ and all pairs $(\mathsf{crs}_{\mathcal{L}}, \tau_{\Sigma}) \leftarrow \mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$, we have $\mathsf{BadChallenge}(\tau_{\Sigma}, \mathsf{crs}, x, \mathbf{a}) = f(\mathsf{crs}, x, \mathbf{a})$.*

Note that the $\mathsf{TrapGen}$ algorithm does not take a specific statement x as input, but only a trapdoor $\tau_{\mathcal{L}}$ allowing to recognize elements of $\mathcal{L}_{\mathsf{sound}}$.

2.7 $\mathcal{R}$-Lossy Public-Key Encryption with Efficient Opening

In [60], Libert *et al.* formalized a generalization of the notion of $\mathcal{R}$-lossy encryption introduced by Boyle *et al.* [19]. The primitive is a tag-based encryption scheme [58] where the tag space $\mathcal{T}$ is partitioned into *injective* tags and *lossy* tags. When ciphertexts are generated for an injective tag, the decryption algorithm correctly recovers the underlying plaintext. When messages are encrypted under lossy tags, the ciphertext is statistically independent of the plaintext. In $\mathcal{R}$-lossy PKE schemes, the tag space is partitioned according to a binary relation $\mathcal{R} \subseteq \mathcal{K} \times \mathcal{T}$. The key generation algorithm takes as input an initialization value $K \in \mathcal{K}$ and partitions $\mathcal{T}$ in such a way that injective tags $t \in \mathcal{T}$ are exactly those for which $(K, t) \in \mathcal{R}$ (i.e., all tags t for which $(K, t) \notin \mathcal{R}$ are lossy).

From a security standpoint, the definitions of [19] require the initialization value K to be computationally hidden by the public key. The definition of [60] requires the existence of a lossy key generation algorithm $\mathsf{LKeygen}$ which outputs public keys with respect to which all tags t are lossy (in contrast with injective keys where the only lossy tags are those for which $(K,t) \notin \mathcal{R}$). In addition, [60] also asks that the secret key allows equivocating lossy ciphertexts (a property called *efficient opening* by Bellare *et al.* [8]) using an algorithm called Opener. For the purpose of constructing simulation-sound arguments, [60] uses two distinct opening algorithms Opener and $\mathsf{LOpener}$. The former operates over injective public keys for lossy tags while the latter can equivocate ciphertexts encrypted under lossy keys for any tag.

Definition 2.22. *Let $\mathcal{R} \subseteq \mathcal{K}_\lambda \times \mathcal{T}_\lambda$ be an efficiently computable binary relation. An $\mathcal{R}$-lossy PKE scheme with efficient opening is a 7-uple of PPT algorithms* $(\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Opener}, \mathsf{LOpener})$ *such that:*

Parameter generation: *On input of a security parameter λ, a desired length of of initialization values $L \in \mathsf{poly}(\lambda)$ and a lower bound $B \in \mathsf{poly}(\lambda)$ on the message length, $\mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$ outputs public parameters Γ that specify a tag space $\mathcal{T}$, a space of initialization values $\mathcal{K}$, a public key space $\mathcal{PK}$, a secret key space $\mathcal{SK}$ and a trapdoor space $\mathcal{TK}$.*

Key generation: *For an initialization value $K \in \mathcal{K}$ and public parameters Γ, algorithm $\mathsf{Keygen}(\Gamma, K)$ outputs an injective public key $\mathsf{pk} \in \mathcal{PK}$, a decryption key $\mathsf{sk} \in \mathcal{SK}$ and a trapdoor key $\mathsf{tk} \in \mathcal{TK}$. The public key specifies a ciphertext space CtSp and a randomness space R^{LPKE}.*

Lossy Key generation: *Given an initialization value $K \in \mathcal{K}$ and public parameters Γ, the lossy key generation algorithm $\mathsf{LKeygen}(\Gamma, K)$ outputs a lossy public key $\mathsf{pk} \in \mathcal{PK}$, a lossy secret key $\mathsf{sk} \in \mathcal{SK}$ and a trapdoor key $\mathsf{tk} \in \mathcal{TK}$.*

Decryption under injective tags: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, any initialization value $K \in \mathcal{K}$, any tag $t \in \mathcal{T}$ such that $(K,t) \in \mathcal{R}$, and any message $\mathsf{Msg} \in \mathsf{MsgSp}$, we have*

$$\Pr\left[\exists r \in R^{\mathsf{LPKE}} : \mathsf{Decrypt}\big(\mathsf{sk}, t, \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}; r)\big) \neq \mathsf{Msg}\right] < \nu(\lambda) \ ,$$

for some negligible function $\nu(\lambda)$, where $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$ and the probability is taken over the randomness of Keygen.

Indistinguishability: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, the key generation algorithms* $\mathsf{LKeygen}$ *and* Keygen *satisfy the following:*

(i) For any $K \in \mathcal{K}$, the distributions $D_{\text{inj}} = \{(\mathsf{pk}, \mathsf{tk}) \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)\}$ and $D_{\text{loss}} = \{(\mathsf{pk}, \mathsf{tk}) \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$ are computationally indistinguishable. Namely, for any PPT adversary $\mathcal{A}$, we have $\mathbf{Adv}^{\mathsf{indist\text{-}LPKE}}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda)$, where

$$\begin{aligned}\mathbf{Adv}^{\mathsf{indist\text{-}LPKE}}_{\mathcal{A}}(\lambda) := \ & \big|\Pr[(\mathsf{pk}, \mathsf{tk}) \hookleftarrow D_{\text{inj}} : \mathcal{A}(\mathsf{pk}, \mathsf{tk}) = 1] \\ & - \Pr[(\mathsf{pk}, \mathsf{tk}) \hookleftarrow D_{\text{loss}} : \mathcal{A}(\mathsf{pk}, \mathsf{tk}) = 1]\big| \ .\end{aligned}$$

(ii) For any initialization values $K, K' \in \mathcal{K}$, the two distributions $\{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$ and $\{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K')\}$

are statistically indistinguishable. We require them to be $2^{-\Omega(\lambda)}$*-close in terms of statistical distance.*

Lossiness: *For any* $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$*, any initialization value* $K \in \mathcal{K}$ *and tag* $t \in \mathcal{T}$ *such that* $(K,t) \notin \mathcal{R}$*, any* $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$*, and any* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$*, the following distributions are statistically close:*

$$\{C \mid C \leftarrow \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_0)\} \quad \approx_s \quad \{C \mid C \leftarrow \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_1)\}.$$

For any $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$*, the above holds for any tag* t *(and not only those for which* $(K,t) \notin \mathcal{R}$*).*

Equivocation under lossy tags: *For any* $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$*, any* $K \in \mathcal{K}$*, any keys* $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$ *let* D_R *denote the distribution, defined over the randomness space* R^{LPKE}*, from which the random coins used by* Encrypt *are sampled. For any message* $\mathsf{Msg} \in \mathsf{MsgSp}$ *and ciphertext* C*, let* $D_{\mathsf{pk},\mathsf{Msg},C,t}$ *denote the probability distribution on* R^{LPKE} *with support*

$$S_{\mathsf{pk},\mathsf{Msg},C,t} = \{\overline{r} \in R^{\mathsf{LPKE}} \mid \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}, \overline{r}) = C\} \ ,$$

and such that, for each $\overline{r} \in S_{PK,\mathsf{Msg},C,t}$*, we have*

$$D_{\mathsf{pk},\mathsf{Msg},C,t}(\overline{r}) = \Pr_{r' \hookleftarrow D_R}[r' = \overline{r} \mid \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}, r') = C] \ . \tag{1}$$

For any random coins $r \hookleftarrow D_R$*, any tag* $t \in \mathcal{T}_\lambda$ *such that* $(K,t) \notin \mathcal{R}$*, and any messages* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$*, algorithm* Opener *takes as inputs* $\mathsf{pk}, C = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_0, r)$*,* r t*, and* tk*. It outputs a sample* $\overline{r}$ *from a distribution statistically close to* $D_{\mathsf{pk},\mathsf{Msg}_1,C,t}$*.*

Equivocation under lossy keys: *For any initialization value* $K \in \mathcal{K}_\lambda$*, any keys* $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$*, any random coins* $r \hookleftarrow D_R$*, any tag* $t \in \mathcal{T}_\lambda$*, and any distinct messages* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$*, algorithm* LOpener *takes as input* $C = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_0, r)$*,* r*,* t *and* sk*. It outputs* $\overline{r} \in R^{\mathsf{LPKE}}$ *such that* $C = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_1, \overline{r})$*. We require that, for any* $t \in \mathcal{T}_\lambda$ *such that* $(K,t) \notin \mathcal{R}$*, the distributions*

$$\{\overline{r} \leftarrow \mathsf{LOpener}(\mathsf{pk}, \mathsf{sk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, r) \mid r \hookleftarrow D_R\}$$

and $\{\overline{r} \leftarrow \mathsf{Opener}(\mathsf{pk}, \mathsf{tk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, r) \mid r \hookleftarrow D_R\}$ *be statistically close.*

The above definition is slightly weaker than the one of [60] in the property of equivocation under lossy keys. Here, we do not require that the outputs of Opener and LOpener be statistically close to $D_{\mathsf{pk},\mathsf{Msg}_1,C,t}$ as defined in (1): We only require that, on lossy keys and lossy tags, Opener and LOpener sample random coins from statistically close distributions. In fact, the first indistinguishability property implies (since the distinguisher is given tk) that the outputs of both algorithms will be *computationally* indistinguishable from $D_{\mathsf{pk},\mathsf{Msg}_1,C,t}$. Our definition turns out to be sufficient for the purpose of simulation-sound arguments and will allow us to obtain a construction from the DCR assumption.

We note that the property of decryption under injective tags does not assume that random coins are honestly sampled, but only that they belong to some predefined set R^{LPKE}.

2.8 Trapdoor Σ-Protocol Showing Composite Residuosity

We recall the trapdoor Σ-protocols of [61], which allows proving that an element of $\mathbb{Z}_{N^2}^*$ is a composite residue (i.e., a Paillier encryption of 0).

Namely, let $N = pq$ be an RSA modulus and let an integer $\zeta > 1$. We describe a trapdoor Σ-protocol for the language

$$\mathcal{L}^{\mathsf{DCR}} := \{x \in \mathbb{Z}_{N^{\zeta+1}}^* \mid \exists w \in \mathbb{Z}_N^\star : x = w^{N^\zeta} \bmod N^{\zeta+1}\}.$$

We assume that the challenge space is $\{0, \ldots, 2^\lambda - 1\}$ and that $p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > \lambda$ for any sufficiently large $\lambda \in \mathbb{N}$. The condition $p, q > 2^\lambda$ will ensure that the difference between any two challenges be co-prime with N.

In order to obtain a $\mathsf{BadChallenge}$ function that identifies bad challenges for elements $x \notin \mathcal{L}^{\mathsf{DCR}}$, one difficulty is the case of elements $x \in \mathbb{Z}_{N^{\zeta+1}}^*$ that are encryptions of an element $\alpha_x \in \mathbb{Z}_N$ such that $1 < \gcd(\alpha_x, N^\zeta) < N^\zeta$. Indeed, we cannot immediately identify a unique bad challenge by inverting α_x in $\mathbb{Z}_{N^\zeta}$. However, a closer analysis shows that, even when $\zeta > 1$ and $\gcd(\alpha_x, N^\zeta) > 1$, at most one bad challenge can exist in the set $\{0, 1, \ldots, 2^\lambda - 1\}$.

$\mathbf{Gen}_{\mathsf{par}}(1^\lambda)$: Given the security parameter λ, define $\mathsf{par} = \{\lambda\}$.

$\mathbf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}^{\mathsf{DCR}})$: Given public parameters par as well as a description of a language $\mathcal{L}^{\mathsf{DCR}}$, consisting of an RSA modulus $N = pq$ with p and q prime satisfying $p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > \lambda$, define the language-dependent $\mathsf{crs}_{\mathcal{L}} = \{N\}$. The global CRS is

$$\mathsf{crs} = (\{\lambda\}, \mathsf{crs}_{\mathcal{L}}).$$

$\mathbf{TrapGen}(\mathsf{par}, \mathcal{L}^{\mathsf{DCR}}, \tau_{\mathcal{L}})$: Given par, the description of a language $\mathcal{L}^{\mathsf{DCR}}$ that specifies an RSA modulus N and a membership-testing trapdoor $\tau_{\mathcal{L}} = (p, q)$ consisting of the factorization of $N = pq$, output the language-dependent $\mathsf{crs}_{\mathcal{L}} = \{N\}$ which defines $\mathsf{crs} = (\{\lambda\}, \mathsf{crs}_{\mathcal{L}})$ and the trapdoor $\tau_\Sigma = (p, q)$.

$\mathbf{P}\big(\mathsf{crs}, x, w\big) \leftrightarrow \mathbf{V}(\mathsf{crs}, x)$: Given a crs, a statement $x = w^{N^\zeta} \bmod N^{\zeta+1}$, P (who has the witness $w \in \mathbb{Z}_N^\star$) and V interact as follows:

1. P chooses a random $r \hookleftarrow U(\mathbb{Z}_N^*)$ and sends $a = r^{N^\zeta} \bmod N^{\zeta+1}$ to V.
2. V sends a random challenge $\mathsf{Chall} \hookleftarrow U(\{0, \ldots, 2^\lambda - 1\})$ to P.
3. P computes the response $z = r \cdot w^{\mathsf{Chall}} \bmod N$ and sends it to V.
4. V checks if $a \cdot x^{\mathsf{Chall}} \equiv z^{N^\zeta} \pmod{N^{\zeta+1}}$ and returns 0 if this condition is not satisfied.

$\mathbf{BadChallenge}\big(\mathsf{par}, \tau_\Sigma, \mathsf{crs}, x, a\big)$: Given $\tau_\Sigma = (p, q)$, decrypt x and a to obtain $\alpha_x = \mathcal{D}_{\tau_\Sigma}(x) \in \mathbb{Z}_{N^\zeta}$, $\alpha_a = \mathcal{D}_{\tau_\Sigma}(a) \in \mathbb{Z}_{N^\zeta}$.

1. If $\alpha_a = 0$, return $\mathsf{Chall} = 0$.
2. If $\alpha_a \neq 0$, let $d_x = \gcd(\alpha_x, N^\zeta)$, which lives in the set

$$\{p^i q^j \mid 0 \leq i < \zeta,\ 0 \leq j < \zeta\} \cup \{p^i q^\zeta \mid 0 \leq i < \zeta\} \cup \{p^s q^j \mid 0 \leq j < \zeta\}.$$

Then, do the following:

a. If $1 < d_x < N^\zeta$, return $\perp$ if d_x does not divide $N^\zeta - \alpha_a$.
b. Otherwise, the congruence $\alpha_a + \mathsf{Chall} \cdot \alpha_x \equiv 0 \pmod{\frac{N^\zeta}{d_x}}$ has a unique solution $\mathsf{Chall}' = -\alpha_x^{-1} \cdot \alpha_a \in \mathbb{Z}_{N^\zeta/d_x}$ since $\gcd(\alpha_x, N^\zeta/d_x) = 1$. If $\mathsf{Chall}' \in \mathbb{Z}_{N^\zeta/d_x} \setminus \{0, \ldots, 2^\lambda - 1\}$, return $\perp$. Else, return $\mathsf{Chall} = \mathsf{Chall}'$.

In [61], it is shown that the above construction is a trapdoor Σ-protocol with large challenge space. By applying [72], this implies compact NIZK arguments (i.e., without using parallel repetitions to achieve negligible soundness error) for the language $\mathcal{L}^{\mathsf{DCR}}$ assuming that the LWE assumption holds.

Lemma 2.23 ([61]). *The above protocol is a trapdoor Σ-protocol for the language $\mathcal{L}^{\mathsf{DCR}}$.*

3 NIZK Arguments with One-Time Simulation-Soundness

Libert *et al.* [60] gave a method that directly compiles (i.e., without relying on generic NIZK techniques [37]) any trapdoor Σ-protocol for a trapdoor language into an unbounded simulation-sound NIZK argument for the *same* language. As a building block, their construction uses an LWE-based equivocable $\mathcal{R}$-lossy PKE scheme for the bit-matching relation. Under the DCR assumption, a more efficient $\mathcal{R}$-lossy PKE scheme was described in [61]. In this section, we show that, in applications that only require *one-time* simulation-soundness, we can use an $\mathcal{R}$-lossy PKE scheme with a constant-size public key. In contrast, the $\mathcal{R}$-lossy PKE system of [61] has a large public key comprised of $\Theta(\lambda)$ Paillier ciphertexts.

In our *one-time* simulation-sound arguments, we use an $\mathcal{R}$-lossy PKE scheme for the inequality relation.

Definition 3.1. *Let $\mathcal{K} = \{0,1\}^\ell$ and $\mathcal{T} = \{0,1\}^\ell$, for some $\ell \in \mathsf{poly}(\lambda)$. The* **inequality relation** *$\mathcal{R}_{\mathsf{NEQ}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$ is the relation where $\mathcal{R}_{\mathsf{NEQ}}(K,t) = 1$ if and only if $K \neq t$.*

3.1 An $\mathcal{R}_{\mathsf{NEQ}}$-Lossy PKE Scheme from DCR

Our DCR-based $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme goes as follows.

Par-Gen$(1^\lambda, 1^L, 1^B)$**:** Define $\mathcal{K} = \mathcal{T} = \{0,1\}^L$, so that the tag and initialization value spaces coincide. Define public parameters as $\Gamma = (1^\lambda, 1^L, 1^B)$.

Keygen(Γ, K)**:** On input of public parameters Γ and $K \in \mathcal{K}$, generate a key pair as follows.

1. Choose an RSA modulus $N = pq$ such that $p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > L(\lambda)$ for any sufficiently large λ, and an integer $\zeta \in \mathsf{poly}(\lambda)$ such that $N^\zeta > 2^B$.
2. Pick $u \hookleftarrow U(\mathbb{Z}^*_{N^{\zeta+1}})$, $\bar{v} \hookleftarrow U(\mathbb{Z}^*_N)$ and compute $v = u^{-K} \cdot \bar{v}^{N^\zeta} \bmod N^{\zeta+1}$, where K is interpreted as an element of $\mathbb{Z}_{N^\zeta}$.

Define $R^{\mathsf{LPKE}} = \mathbb{Z}_N^*$ and output $\mathsf{sk} = (p, q, K)$ as well as

$$\mathsf{pk} := \Big(N, \zeta, u, v\Big), \qquad \mathsf{tk} = (\bar{v}, K).$$

LKeygen(Γ, K)**:** On input of public parameters Γ and an initialization value $K \in \mathcal{K}$, generate a key pair as follows.

1. Choose an RSA modulus $N = pq$ such that $p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > L(\lambda)$ for any sufficiently large λ, and an integer $\zeta \in \mathsf{poly}(\lambda)$ such that $N^\zeta > 2^B$.
2. Choose $\bar{u}, \bar{v} \hookleftarrow U(\mathbb{Z}_N^*)$ uniformly. Compute $u = \bar{u}^{N^\zeta} \bmod N^{\zeta+1}$ and $v = u^{-K} \cdot \bar{v}^{N^\zeta} \bmod N^{\zeta+1}$, where K is interpreted as an element of $\mathbb{Z}_{N^\zeta}$.

Define $R^{\mathsf{LPKE}} = \mathbb{Z}_N^*$ and output $\mathsf{sk} = (\bar{u}, \bar{v}, K)$ as well as $\mathsf{pk} := \big(N, \zeta, u, v\big)$ and $\mathsf{tk} = (\bar{v}, K)$.

Encrypt$(\mathsf{pk}, t, \mathsf{Msg})$**:** To encrypt $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$ for the tag $t \in \{0,1\}^L$, interpret t as an element of $\mathbb{Z}_{N^\zeta}$. Pick $r \hookleftarrow U(\mathbb{Z}_N^*)$ and compute

$$\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}} \cdot r^{N^\zeta} \bmod N^{\zeta+1}.$$

Decrypt$(\mathsf{sk}, t, \mathsf{ct})$**:** Given $sk = (p, q, t^\star)$ and the tag $t \in \{0,1\}^L$, interpret t as an element of $\mathbb{Z}_{N^\zeta}$. Then, do the following:

1. Letting $\lambda(N) = \mathrm{lcm}(p-1, q-1)$, compute $h_t = (u^t \cdot v)^{\lambda(N)} \bmod N^{\zeta+1}$, which can be written $h_t = 1 + g_t N \bmod N^{\zeta+1}$, for some $g_t \in \mathbb{Z}_{N^\zeta}$, since its order is at most N^ζ. Return $\perp$ if $g_t = 0$ or $\gcd(g_t, N^\zeta) > 1$.
2. Otherwise, compute $\mathsf{Msg} = \frac{(\mathsf{ct}^{\lambda(N)} \bmod N^{\zeta+1}) - 1}{N} \cdot g_t^{-1} \bmod N^\zeta$, where the division is computed over $\mathbb{Z}$, and output $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$.

Opener$(\mathsf{pk}, \mathsf{tk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, r)$**:** Given $\mathsf{tk} = (\bar{v}, K)$ and $t \in \{0,1\}^L$, return $\perp$ if $t \neq K$ when they are interpreted as elements of $\mathbb{Z}_{N^\zeta}$. Otherwise, given $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathbb{Z}_{N^\zeta}$ and $r \in \mathbb{Z}_N^*$ such that

$$\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}_0} \cdot r^{N^\zeta} = (\bar{v}^N)^{\mathsf{Msg}_0} \cdot r^{N^\zeta} \bmod N^{\zeta+1}, \tag{2}$$

output $\bar{r} = r \cdot \bar{v}^{\mathsf{Msg}_0 - \mathsf{Msg}_1} \bmod N$, so that $\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}_1} \cdot \bar{r}^{N^\zeta} \bmod N^{\zeta+1}$.

LOpener$(\mathsf{sk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, r)$**:** Given $\mathsf{sk} = (\bar{u}, \bar{v}, K)$ and $t \in \{0,1\}^L$, interpret t as an element of $\mathbb{Z}_{N^\zeta}$. Given $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathbb{Z}_{N^\zeta}$ and $r \in \mathbb{Z}_N^*$ such that

$$\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}_0} \cdot r^{N^\zeta} = (\bar{u}^{t-K} \cdot \bar{v})^{N \cdot \mathsf{Msg}_0} \cdot r^{N^\zeta} \bmod N^{\zeta+1}, \tag{3}$$

output $\bar{r} = r \cdot (\bar{u}^{t-K} \cdot \bar{v})^{\mathsf{Msg}_0 - \mathsf{Msg}_1} \bmod N$, which satisfies

$$\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}_1} \cdot \bar{r}^{N^\zeta} \bmod N^{\zeta+1}.$$

The scheme enables decryption under injective tags because, with high probability over the randomness of Keygen, the order of u is a multiple of N^ζ since u is sampled uniformly in $\mathbb{Z}_{N^{\zeta+1}}^*$ at step 2. Since $t, K \in \{0,1\}^L$ and $p, q > 2^L$, we have $\gcd(t - K, N^\zeta) = 1$, so that N^ζ divides the order of $u^{t-K} \cdot \bar{v}^{N^\zeta}$ when $t \neq K$. This ensures that h_t has order N^ζ and $\gcd(g_t, N^\zeta) = 1$ at step 1 of Decrypt.

We now prove that the scheme satisfies all the properties of Definition 2.22. The first indistinguishability property crucially imposes that lossy and injective keys be indistinguishable even when the equivocation trapdoor tk of Opener is given. This is important for our proof of one-time simulation-soundness, which requires that Opener be able to equivocate lossy ciphertexts given only tk and without knowing the factorization of N (otherwise, we could not meaningfully rely on the DCR assumption to switch from lossy to injective keys).

Theorem 3.2. *The above construction is an $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme under the* DCR *assumption.* (The proof is available in the full version of the paper [40]).

3.2 The Argument System

Our one-time simulation-sound argument is very similar to the one of [60] which provides unbounded simulation-soundness using a more expensive $\mathcal{R}$-lossy PKE scheme. The construction relies on the following ingredients.

- A trapdoor Σ-protocol $\Pi' = (\mathsf{Gen}'_{\mathsf{par}}, \mathsf{Gen}'_{\mathcal{L}}, \mathsf{P}', \mathsf{V}')$ for an NP language $\mathcal{L}$. This protocol should satisfy the properties of Definition 2.21. In addition, the function $\mathsf{BadChallenge}(\tau_\Sigma, \mathsf{crs}, x, a)$ should be computable within time $T \in \mathsf{poly}(\lambda)$ for any input $(\tau, \mathsf{crs}, x, a)$. Let also $B \in \mathsf{poly}(\lambda)$ the maximal length of the first prover message sent by P'.
- A strongly unforgeable one-time signature scheme $\mathsf{OTS} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys in $\{0,1\}^L$, where $L \in \mathsf{poly}(\lambda)$.
- An $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme $\Pi^{\mathsf{LPKE}} = (\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Opener}, \mathsf{LOpener})$ with space $\mathcal{K} = \mathcal{T} = \{0,1\}^L$. We assume that its decryption algorithm is computable within time T.
- A correlation intractable hash family $\mathcal{H} = (\mathsf{Gen}, \mathsf{Hash})$ for the class $\mathcal{R}_{\mathsf{CI}}$ of relations that are efficiently searchable within time T.

$\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$**:** Run $\mathsf{par} \leftarrow \mathsf{Gen}'_{\mathsf{par}}(1^\lambda)$ and output par.

$\mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$**:** Given public parameters par and a language $\mathcal{L}$, the CRS is generated as follows.

1. Generate a CRS $\mathsf{crs}'_{\mathcal{L}} \leftarrow \mathsf{Gen}'_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$ for the trapdoor Σ-protocol Π'.
2. Choose the description a one-time signature scheme $\mathsf{OTS} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys in $\{0,1\}^L$, where $L \in \mathsf{poly}(\lambda)$.
3. Choose public parameters $\Gamma \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$ for an $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme with tag space $\mathcal{K} = \mathcal{T} = \{0,1\}^L$. Then, generate lossy keys $(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{sk}_{\mathsf{LPKE}}, \mathsf{tk}_{\mathsf{LPKE}}) \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{LKeygen}(\Gamma, 0^L)$.
4. Generate a key $k \leftarrow \mathsf{Gen}(1^\lambda)$ for a correlation intractable hash function with output length $\kappa = \Theta(\lambda)$.

Output the language-dependent CRS $\mathsf{crs}_{\mathcal{L}} := \big(\mathsf{crs}'_{\mathcal{L}}, \mathsf{pk}_{\mathsf{LPKE}}, k\big)$ and the simulation trapdoor $\tau_{\mathsf{zk}} := \mathsf{sk}_{\mathsf{LPKE}}$. The global common reference string consists of $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}}, \mathsf{pk}_{\mathsf{LPKE}}, \mathsf{OTS})$.

$\mathbf{P}(\mathsf{crs}, x, w, \mathsf{lbl})$: To prove a statement $x \in \mathcal{L}$ for a label $\mathsf{lbl} \in \{0,1\}^*$ using the witness w, generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(1^\lambda)$. Then,

1. Compute $(a', st') \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w)$. Then, sample $r \hookleftarrow D_R^{\mathsf{LPKE}}$ in the randomness space R^{LPKE} of Π^{LPKE}. Using the tag $\mathsf{VK} \in \{0,1\}^L$, compute $a \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{Encrypt}(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{VK}, a'; r)$.
2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, a, \mathsf{VK}))$.
3. Compute $z' = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, a', \mathsf{Chall}, st')$ by executing the prover of Π'. Define $z = (z', a', r)$.
4. Generate $sig \leftarrow \mathcal{S}(\mathsf{SK}, (x, a, z, \mathsf{lbl}))$ and output $\boldsymbol{\pi} = \big(\mathsf{VK}, (a, z), sig\big)$.

$\mathbf{V}(\mathsf{crs}, x, \boldsymbol{\pi}, \mathsf{lbl})$: Given a statement x, a label lbl as well as a purported proof $\boldsymbol{\pi} = \big(\mathsf{VK}, (a, z), sig\big)$, return 0 if $\mathcal{V}(\mathsf{VK}, (x, a, z, \mathsf{lbl}), sig) = 0$. Otherwise,

1. Write $z = (z', a', r)$ and return 0 if any of these does not parse properly or if $a \neq \Pi^{\mathsf{LPKE}}.\mathsf{Encrypt}(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{VK}, a'; r)$.
2. Let $\mathsf{Chall} = \mathsf{Hash}\big(k, (x, a, \mathsf{VK})\big)$. If $\mathsf{V}'(\mathsf{crs}'_{\mathcal{L}}, x, a', \mathsf{Chall}, z')) = 1$, return 1. Otherwise, return 0.

Theorem 3.3. *The above argument is statistically (resp. computationally) zero-knowledge if: (i) Π^{LPKE} is statistically equivocable under lossy keys; (ii) The trapdoor Σ-protocol Π' is statistically (resp. computationally) special zero-knowledge.* (The proof is given in the full version of the paper.)

Theorem 3.4. *The above construction provides one-time simulation-soundness if: (i) OTS is a strongly unforgeable one-time signature; (ii) Π^{LPKE} is an $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme; (iii) The hash function family $\mathcal{H}$ is correlation-intractable for all relations that are searchable within time T, where T denotes the maximal running time of algorithms $\mathsf{BadChallenge}(\cdot,\cdot,\cdot,\cdot)$ and $\Pi^{\mathsf{LPKE}}.\mathsf{Decrypt}(\cdot,\cdot,\cdot)$.* (The proof is given in the full version of the paper.)

4 An Adaptively Secure CCA2-Secure Threshold Encryption Scheme Based on Paillier and LWE

Our construction combines a one-time simulation-sound argument of composite residuosity with a threshold variant of an Elgamal-Paillier combination due to Camenisch and Shoup [20]. As in [66], we use a generalization of the Camenisch-Shoup system based on ideas from Damgård and Jurik [34].

For simplicity, we first present a non-robust version of the scheme. In the full version of the paper, we explain how to obtain robustness against malicious adversaries by having each server prove that its decryption share is consistent with some public commitment to its corresponding secret key share.

$\mathsf{KeyGen}(1^\lambda, \mathbb{A})$: The dealer conducts the following steps:

1. Choose a safe-prime product $N = pq$, of which the prime factors are of the form $p = 2p' + 1$, $q = 2q' + 1$ for some primes primes $p', q' > 2^{l(\lambda)}$, where $l : \mathbb{N} \to \mathbb{N}$ is a polynomial. Choose an integer $\zeta \geq 1$ so as to define the message space as $\mathsf{MsgSp} = \mathbb{Z}_{N^\zeta}$. Then, define the language

$$\mathcal{L}^{\mathsf{DCR}} := \{x \in \mathbb{Z}^*_{N^{\zeta+1}} \mid \exists w \in \mathbb{Z}^\star_N : x = w^{N^\zeta} \bmod N^{\zeta+1}\}$$

and choose $g_0 \hookleftarrow U(\mathbb{Z}^*_N)$.

2. Generate a common reference string $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ for the one-time simulation-sound argument system $\Pi^{\mathsf{OTSS}} = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ of Sect. 3.
3. Let $\sigma > \sqrt{\lambda \cdot e} \cdot N^\zeta$ be a Gaussian parameter, where $e = \Omega(\ell^{(1+\sqrt{2})/2})$ is the dimension of the matrix $\mathbf{M}$ in Sect. 2.3. Sample a secret key $x \hookleftarrow D_{\mathbb{Z},\sigma}$ and compute $h = g_0^{4N^\zeta \cdot x} \bmod N^{\zeta+1}$. Define the public key $\mathsf{pk} := (N, \zeta, g_0, \mathsf{crs})$ whereas the centralized secret key $\mathsf{sk} := x \in \mathbb{Z}$.
4. Share sk using a LISS scheme. To this end, sample $\bar{\boldsymbol{\rho}} = (\rho_2, \ldots, \rho_e)^\top \hookleftarrow (D_{\mathbb{Z},\sigma})^{(e-1)}$, define $\boldsymbol{\rho} = [x \mid \bar{\boldsymbol{\rho}}^\top]^\top \in \mathbb{Z}^e$ and compute

$$\boldsymbol{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_d \end{bmatrix} = \mathbf{M} \cdot \boldsymbol{\rho} \in \mathbb{Z}^d,$$

where $\mathbf{M} \in \mathbb{Z}^{d \times e}$ is the share-generating matrix of Sect. 2.3, which computes the Boolean formula associated with the threshold access structure $\mathbb{A}$. Then, define the private key shares as

$$\mathsf{sk}_i = \big(s_j\big)_{j \in \psi^{-1}(i)} = \big(\mathbf{M}_j \cdot \boldsymbol{\rho}\big)_{j \in \psi^{-1}(i)} \in \mathbb{Z}^{d_i} \qquad \forall i \in [\ell],$$

where $\mathbf{M}_j \in \mathbb{Z}^{1 \times e}$ denotes the j-th row of $\mathbf{M}$ while d_i stands for the number of rows assigned by the LISS scheme to server i.

Finally, output the public key $\mathsf{pk} = (N, \zeta, g_0, \mathsf{crs})$ and the vector of secret-key shares $(\mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell)$.

Encrypt$(\mathsf{pp}, \mathsf{pk}, \mathsf{Msg})$**:** To encrypt $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$, choose $r \hookleftarrow U(\{0, \ldots, \lfloor N/4 \rfloor\})$ and compute

$$C_0 = {g_0}^{2N^\zeta \cdot r} \bmod N^{\zeta+1} \qquad C_1 = (1+N)^{\mathsf{Msg}} \cdot h^r \bmod N^{\zeta+1}$$

Then, using the witness $w = g_0^{2r} \bmod N$, compute a simulation-sound NIZK argument $\boldsymbol{\pi} \leftarrow \mathsf{P}\big(\mathsf{crs}, C_0, g_0^{2r} \bmod N, \mathsf{lbl}\big)$ that $C_0 \in \mathcal{L}^{\mathsf{DCR}}$ using the label $\mathsf{lbl} = C_1$. Then, return the ciphertext $\mathsf{ct} := (C_0, C_1, \boldsymbol{\pi})$.

PartDec$\big(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct}\big)$**:** On input of its share $\mathsf{sk}_i = \{s_j = \mathbf{M}_j \cdot \boldsymbol{\rho}\}_{j \in \psi^{-1}(i)}$ and a ciphertext $\mathsf{ct} = (C_0, C_1, \boldsymbol{\pi})$, the i-th server does the following.

1. If $\mathsf{V}(\mathsf{crs}, C_0, \boldsymbol{\pi}, \mathsf{lbl}) = 0$, return $\perp$.
2. For each $j \in \psi^{-1}(i) = \{j_1, \ldots, j_{d_i}\}$, compute $\mu_{i,j} = C_0^{2 \cdot s_j} \bmod N^{\zeta+1}$ and return

$$\begin{aligned} \boldsymbol{\mu}_i &= (\mu_{i,j_1}, \ldots, \mu_{i,j_{d_i}}) \\ &= \big(C_0^{2 \cdot s_{j_1}} \bmod N^{\zeta+1},\ \ldots,\ C_0^{2 \cdot s_{j_{d_i}}} \bmod N^{\zeta+1}\big) \in (\mathbb{Z}^*_{N^{\zeta+1}})^{d_i}. \end{aligned}$$

Combine$\big(\mathsf{pp}, \mathcal{B} = (\mathcal{S} \in \mathbb{A}, \{\boldsymbol{\mu}_i\}_{i \in \mathcal{S}}), \mathsf{ct} = (C_0, C_1, \boldsymbol{\pi})\big)$**:** First, parse the set $\mathcal{S}$ as $\mathcal{S} = \{j_1, \ldots, j_t\}$ and find a vector $\boldsymbol{\lambda}_\mathcal{S} = [\boldsymbol{\lambda}_{j_1}^\top \mid \ldots \mid \boldsymbol{\lambda}_{j_t}^\top]^\top \in \{-1, 0, 1\}^{d_\mathcal{S}}$ such that $\boldsymbol{\lambda}_\mathcal{S} \cdot \mathbf{M}_{\psi^{-1}(\mathcal{S})} = (1, 0, \ldots, 0)$, where $d_\mathcal{S} = \sum_{i \in \mathcal{S}} d_i$ and $\boldsymbol{\lambda}_{j_i} = (\lambda_{j_i,1}, \ldots, \lambda_{j_i,d_{j_i}}) \in \{-1, 0, 1\}^{d_i}$ for all $i \in [t]$. Then, do the following:

1. Compute
$$\hat{\mu} \triangleq \prod_{i\in[t]} \prod_{k\in[d_{j_i}]} \mu_{j_i,k}^{\lambda_{j_i,k}} \bmod N^{\zeta+1}.$$
2. Compute $\hat{C}_1 = C_1/\hat{\mu} \bmod N^{\zeta+1}$ and return $\perp$ if $\hat{C}_1 \not\equiv 1 \pmod N$. Otherwise, return $\mathsf{Msg} = (\hat{C}_1 - 1)/N \in \mathbb{Z}_{N^\zeta}$.

In the dealing phase, the matrix $\mathbf{M} \in \mathbb{Z}^{d\times e}$ has $O(\log \ell)$ non-zero entries for threshold access structures. If we apply the LISS scheme based on the Benaloh-Leichter secret sharing [9] and the result of Hoory *et al.* [54], $\mathbf{M}$ has dimensions $d, e = O(\ell^{1+\sqrt{2}})$, so that its rows have norm $\|\mathbf{M}_j\| = O(\sqrt{e}\log \ell)$, which leads to share units of magnitude $|s_j| = O(\sigma e \cdot \log \ell)$.

The scheme thus provides compactness in the sense of Definition 2.13 since the size of ciphertexts and public keys only depends on λ. By increasing the exponent $\zeta > 1$, the ratio between ciphertext and plaintext sizes can approach 1, which was not possible in [62,65].[6] We now prove security in the sense of Definition 2.16.

Theorem 4.1. *The above scheme provides IND-CCA security in the adaptive corruption setting assuming that: (i) The* DCR *assumption holds; (ii) The argument system* Π^{OTSS} *provides one-time simulation-soundness.*

Proof. We consider a sequence of games where, for each i, we call W_i the event that the adversary wins in Game_i.

Game_0: This is the real IND-CCA game. The challenger faithfully answers all queries. In the challenge phase, the adversary $\mathcal{A}$ chooses two messages $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathbb{Z}_{N^\zeta}$. The challenger flips a coin $b \hookleftarrow U(\{0,1\})$ and computes the challenge ciphertext $\mathsf{ct}^* = (C_0^*, C_1^*, \boldsymbol{\pi}^\star)$ by running the real encryption algorithm. When $\mathcal{A}$ halts, it outputs $b' \in \{0,1\}$ and we denote by W_0 the event that $b' = b$. By definition $\mathbf{Adv}^{\text{ind-cca}}(\mathcal{A}) := |\Pr[W_0] - 1/2|$.

Game_1: This game is identical to Game_0 except that we change the generation of the common reference string and the generation of $\boldsymbol{\pi}^\star$ in the challenge ciphertext. In the key generation phase, the challenger runs $(\mathsf{crs}, \tau_{\mathsf{zk}}) \leftarrow \mathsf{Sim}_0(1^\lambda, \mathcal{L}^{\mathsf{DCR}})$. In the challenge ciphertext $\mathsf{ct}^\star = (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$, the NIZK argument $\boldsymbol{\pi}^\star$ is simulated as $\boldsymbol{\pi}^\star \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, C_0^\star, C_1^\star)$ without using the witness. From the perfect zero-knowledge property of Π^{OTSS}, Game_1 is indistinguishable from Game_0 and $\Pr[W_1] = \Pr[W_0]$.

Game_2: This game is identical to Game_1 except that we change the generation of the challenge ciphertext $\mathsf{ct}^\star = (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$. Now, the challenger first samples $z_0 \hookleftarrow U(\mathbb{Z}_N^*)$, which is used to compute $z = z_0^{N^\zeta} \bmod N^{\zeta+1}$ and then

$$C_0^\star = z^2 \bmod N^{\zeta+1}, \qquad C_1^\star = (1+N)^{\mathsf{Msg}_b} \cdot C_0^{\star 2x} \bmod N^{\zeta+1}, \tag{4}$$

[6] While the rate can be optimized via hybrid encryption, this would ruin the voting-friendly property of the scheme [12]. Moreover, the KEM/DEM framework does not immediately work in the threshold setting (see, e.g., [3]).

before simulating $\boldsymbol{\pi}^\star \leftarrow \mathsf{Sim}_1\big(\mathsf{crs}, \tau_{\mathsf{zk}}, C_0^\star, C_1^\star\big)$ as in Game_1. Since the subgroup of $2N^\zeta$-th residues is a cyclic group of order $p'q'$, the distribution of $(C_0^\star, C_1^\star)$ is statistically close to that of Game_1. Indeed, the distribution of $C_0^\star$ is now perfectly (instead of statistically) uniform in the subgroup of $2N^\zeta$-th residues. Hence, $|\Pr[W_2] - \Pr[W_1]| < 2^{-\Omega(\lambda)}$.

Game_3: This game is like Game_2 except that, in order to construct the challenge ciphertext, we now sample $z \hookleftarrow U(\mathbb{Z}^\star_{N^{\zeta+1}})$ uniformly in $\mathbb{Z}^*_{N^{\zeta+1}}$ instead of sampling it from the subgroup of N^ζ-th residues. Then, $(C_0^\star, C_1^\star)$ are still computed as per (4). Under the DCR assumption, this change goes unnoticed and a straightforward reduction shows that $|\Pr[W_3] - \Pr[W_2]| \leq \mathbf{Adv}^{\mathsf{DCR}}(\lambda)$.

At this point, we are done with the DCR assumption and we can henceforth use the factorization of N in subsequent games.

Game_4: In this game, the challenger rejects all pre-challenge partial decryption queries $\mathsf{ct} = (C_0, C_1, \boldsymbol{\pi})$ such that C_0 is not an N^ζ-th residue (note that this can be efficiently checked using the factorization of N). The soundness of the argument system (which is implied by its simulation-soundness) implies that the probability to reject a ciphertext that would not have been rejected in Game_3 is negligible: we have $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}^{\mathsf{OTSS}}(\lambda)$.

Game_5: We modify the partial decryption oracle and now reject post-challenge queries $\mathsf{ct} = (C_0, C_1, \boldsymbol{\pi})$ such that $(C_0, C_1, \boldsymbol{\pi}) \neq (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$ and C_0 is not an N^ζ-th residue. By doing so, the challenger does not reject a ciphertext that would not have been rejected in Game_4 until the event F_5 that $\mathcal{A}$ queries the partial decryption of a ciphertext $\mathsf{ct} = (C_0, C_1, \boldsymbol{\pi}) \neq (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$ such that $\mathsf{V}(\mathsf{crs}, C_0, \boldsymbol{\pi}, C_1) = 1$ although $C_0^{2p'q'} \bmod N^{\zeta+1} \neq 1$. Clearly, event F_5 would contradict the one-time simulation-soundness of the NIZK argument system Π^{OTSS}. We have $|\Pr[W_5] - \Pr[W_4]| \leq \mathbf{Adv}^{\mathsf{OTSS}}(\lambda)$.

Game_6: We finally modify the challenge ciphertext and now compute $(C_0^\star, C_1^\star)$ by sampling $C_0^\star \hookleftarrow \mathbb{QR}_{N^{\zeta+1}}$ as a random quadratic residue in $\mathbb{Z}^*_{N^{\zeta+1}}$ and computing $C_1^\star = (1+N)^{\mathsf{Msg}^\star} \cdot C_0^{\star 2x} \bmod N^{\zeta+1}$ for a random $\mathsf{Msg}^\star \hookleftarrow U(\mathbb{Z}_{N^\zeta})$. Lemma 4.2 shows that Game_6 and Game_5 are negligibly far apart in terms of statistical distance, so that $|\Pr[W_6] - \Pr[W_5]| \leq 2^{-\lambda}$.

In Game_6, we have $\Pr[W_6] = 1/2$ since $\mathsf{ct}^\star$ is completely independent of the challenger's bit $b \sim U(\{0,1\})$. □

Lemma 4.2. Game_6 *and* Game_5 *are statistically indistinguishable.*

Proof. The proof uses similar arguments to [4, Theorem 5]. In Game_5, the challenge ciphertext has components $(C_0^\star, C_1^\star)$ of the form

$$C_0^\star = (1+N)^{\alpha_z} \cdot g^{\beta_z} \bmod N^{\zeta+1},$$
$$C_1^\star = (1+N)^{\mathsf{Msg}_b + 2\alpha_z \cdot (x \bmod N^\zeta)} \cdot g^{2\beta_z \cdot (x \bmod p'q')} \bmod N^{\zeta+1},$$

with $g = g_0^{2N^\zeta} \bmod N^{\zeta+1}$ and for uniform $\alpha_z \sim U(\mathbb{Z}_{N^\zeta})$, $\beta_z \sim U(\mathbb{Z}_{p'q'})$. Since $\gcd(2\alpha_z, N^\zeta) = 1$ with overwhelming probability $\varphi(N)/N$, we only need to show

that, from $\mathcal{A}$'s view, $x \bmod N^\zeta$ is statistically uniform over $\mathbb{Z}_{N^\zeta}$ in order to prove that the distribution of $(C_0^\star, C_1^\star)$ is statistically close to that of Game_6.

In Game_5, we note that the challenger rejects all ciphertexts $\mathsf{ct} = (C_0, C_1, \boldsymbol{\pi})$ such that $C_0 \notin \mathcal{L}^{\mathsf{DCR}}$ and $(C_0, C_1, \boldsymbol{\pi}) \neq (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$. For each partial decryption query $(i, (C_0, C_1, \boldsymbol{\pi}))$ such that $C_0 \in \mathcal{L}^{\mathsf{DCR}}$, the adversary can only learn the information $\{\mathbf{M}_j \cdot \boldsymbol{\rho} \bmod p'q'\}_{j \in \psi^{-1}(i)}$. As for partial decryption queries involving the challenge ciphertext $\mathsf{ct}^\star = (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$, we can handle them as if they were corruption queries since the latter reveal at least as much information as the former. Let $\mathcal{C}^\star$ the set of parties for which the adversary made either a corruption query or a decryption query on the challenge $\mathsf{ct}^\star = (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$. Let $\mathbf{M}_{\mathcal{C}^\star}$ to be the sub-matrix of $\mathbf{M}$ obtained by stacking up the rows assigned to those parties.

Since $\mathcal{C}^\star$ is not authorized in $\mathbb{A}$, there exists $\boldsymbol{\kappa} \in \mathbb{Z}^e$ such that $\kappa_1 = 1$ and $\mathbf{M}_{\mathcal{C}^\star} \cdot \boldsymbol{\kappa} = \mathbf{0}^{d_{\mathcal{C}^\star}}$. Let a matrix $\mathbf{L}$ whose rows form a basis of the lattice $\{\mathbf{m} \in \mathbb{Z}^e, \langle \mathbf{m}, \boldsymbol{\kappa} \rangle = 0\}$, where the rows of $\mathbf{M}_{\mathcal{C}^\star}$ live. Note that $(\mathbf{L}, \mathbf{L} \cdot \boldsymbol{\rho})$ reveals at least as much information as $(\mathbf{M}_{\mathcal{C}^\star}, \mathbf{M}_{\mathcal{C}^\star} \cdot \boldsymbol{\rho})$, so that we may condition on $(\mathbf{L}, \mathbf{L} \cdot \boldsymbol{\rho})$. When we additionally condition on $(\mathbf{M}_{[\ell] \setminus \mathcal{C}^\star} \cdot \boldsymbol{\rho} \bmod p'q')$, we condition on something that reveals fewer information than $\boldsymbol{\rho} \bmod p'q'$.

Let an arbitrary vector $\boldsymbol{\rho}_0 \in \mathbb{Z}^e$ satisfying

$$\mathbf{L} \cdot \boldsymbol{\rho}_0 = \mathbf{L} \cdot \boldsymbol{\rho}, \qquad \boldsymbol{\rho}_0 \equiv \boldsymbol{\rho} \pmod{p'q'}.$$

The conditional distribution of $\boldsymbol{\rho}$ is $\boldsymbol{\rho}_0 + D_{\Lambda, \sigma, -\boldsymbol{\rho}_0}$, where

$$\Lambda = \{\boldsymbol{x} \in \mathbb{Z}^e \mid \mathbf{L} \cdot \boldsymbol{x} = 0 \quad \wedge \quad \boldsymbol{x} = \mathbf{0} \bmod p'q'\}$$

is the lattice $\Lambda = \boldsymbol{\kappa} \cdot \mathbb{Z} \cap (p'q' \cdot \mathbb{Z}^e) = (p'q' \cdot \mathbb{Z}) \cdot \boldsymbol{\kappa}$. Let us write $\boldsymbol{\rho}_0 = y \cdot \boldsymbol{\kappa} + (\boldsymbol{\rho}_0^\perp)$, where $y \in \mathbb{R}$ and $\boldsymbol{\rho}_0^\perp \in \mathbb{Z}^e$ is orthogonal to $\boldsymbol{\kappa}$. Conditionally on $\mathbf{L} \cdot \boldsymbol{\rho}$ and $\boldsymbol{\rho} \bmod p'q'$, the distribution of $\boldsymbol{\rho}$ can be written

$$\begin{aligned} \boldsymbol{\rho}_0 + D_{\Lambda, \sigma, -\boldsymbol{\rho}_0} &= (\boldsymbol{\rho}_0^\perp) + y \cdot \boldsymbol{\kappa} + D_{(p'q' \cdot \mathbb{Z}) \cdot \boldsymbol{\kappa}, \sigma, -(\boldsymbol{\rho}_0^\perp) - y \cdot \boldsymbol{\kappa}} \\ &= (\boldsymbol{\rho}_0^\perp) + y \cdot \boldsymbol{\kappa} + \boldsymbol{\kappa} \cdot D_{(p'q' \cdot \mathbb{Z}), \sigma / \|\boldsymbol{\kappa}\|, -y}. \end{aligned}$$

Since $\kappa_1 = 1$, the conditional distribution of $x = \langle (1, 0, \ldots, 0), \boldsymbol{\rho} \rangle$ is thus

$$c + D_{(p'q' \cdot \mathbb{Z}), \sigma / \|\boldsymbol{\kappa}\|, -y},$$

where $c = y + \langle (1, 0, \ldots, 0), \boldsymbol{\rho}_0^\perp \rangle$. We now consider the distribution obtained by reducing the distribution $D_{(p'q' \cdot \mathbb{Z}), \sigma / \|\boldsymbol{\kappa}\|, -y}$ over $\Lambda_0 = p'q' \cdot \mathbb{Z}$ modulo its sublattice $\Lambda_0' = (p'q') \cdot (N^\zeta \mathbb{Z})$. Since $p'q' \cdot N^\zeta < N^{\zeta+1}$, by Lemma 2.7, choosing the standard deviation $\sigma > \sqrt{\lambda \cdot e} \cdot N^{\zeta+1}$ suffices (by [51, Lemma 3.1] which implies $\eta_\epsilon(\Lambda_0') < \lambda^{1/2} N^{\zeta+1}$) to ensure that $x \bmod N^\zeta$ is within distance $2^{-\lambda}$ from $U(\Lambda_0 / \Lambda_0')$ conditionally on $\mathcal{A}$'s view. This completes the proof since $\gcd(p'q', N^\zeta) = 1$ implies $\Lambda_0 / \Lambda_0' \simeq \mathbb{Z}_{N^\zeta}$. □

In the full version of the paper, we show how to turn the scheme into a robust TPKE system. This is achieved by using trapdoor Σ-protocols to prove the validity of decryption shares. To this end, we need to first construct a standard Σ-protocol with binary challenges in order to apply the generic trapdoor

Σ-protocol construction of Ciampi *et al.* [26]. The disadvantage of this approach is that parallel repetitions incur a communication overhead $\Theta(\lambda)$. In applications to voting (where "non-malleable" ciphertext components are removed from ciphertexts before homomorphically processing them), this may be acceptable if proofs of correct partial decryptions are computed by trustees with higher computational resources than voters. It remains an interesting open problem to achieve robustness without parallel repetitions.

Acknowledgements. Part of this research was funded by the French ANR ALAMBIC project (ANR-16-CE39-0006). This work was also supported in part by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701). Khoa Nguyen was supported in part by the Gopalakrishnan - NTU PPF 2018, by A*STAR, Singapore under research grant SERC A19E3b0099, and by Vietnam National University HoChiMinh City (VNU-HCM) under grant number NCM2019-18-01.

References

1. Abe, M.: Robust distributed multiplication without interaction. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 130–147. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_9
2. Abe, M., Fehr, S.: Adaptively secure feldman vss and applications to universally-composable threshold cryptography. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 317–334. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_20
3. Abe, M., Gennaro, R., Kurosawa, K., Shoup, V.: Tag-KEM/DEM: a new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 128–146. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_8
4. Agrawal, S., Libert, B., Stehlé, D.: Fully secure functional encryption for inner products, from standard assumptions. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9816, pp. 333–362. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53015-3_12
5. Almansa, J.F., Damgård, I., Nielsen, J.B.: Simplified threshold RSA with adaptive and proactive security. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 593–611. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_35
6. Alperin-Sheriff, J., Peikert, C.: Circular and KDM security for identity-based encryption. In: PKC (2012)
7. Asharov, G., Jain, A., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. Cryptology ePrint Archive: Report 2011/613 (2012)
8. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_1
9. Benaloh, J., Leichter, J.: Generalized secret sharing and monotone functions. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 27–35. Springer, New York (1990). https://doi.org/10.1007/0-387-34799-2_3

10. Bendlin, R., Damgård, I.: Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 201–218. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_13
11. Bendlin, R., Krehbiel, S., Peikert, C.: How to share a lattice trapdoor: threshold protocols for signatures and (H)IBE. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) ACNS 2013. LNCS, vol. 7954, pp. 218–236. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38980-1_14
12. Bernhard, D., Cortier, V., Pereira, O., Smyth, B., Warinschi, B.: Adapting helios for provable ballot privacy. In: Atluri, V., Diaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 335–354. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23822-2_19
13. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_27
14. Boneh, D., Boyen, X., Halevi, S.: Chosen ciphertext secure public key threshold encryption without random oracles. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 226–243. Springer, Heidelberg (2006). https://doi.org/10.1007/11605805_15
15. Boneh, D., et al.: Threshold cryptosystems from threshold fully homomorphic encryption. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10991, pp. 565–596. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96884-1_19
16. Boneh, D., Lewi, K., Montgomery, H., Raghunathan, A.: Key homomorphic PRFs and their applications. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 410–428. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_23
17. Boyd, C.: Digital multisignatures. In: Cryptography and Coding (1989)
18. Boyen, X., Mei, Q., Waters, B.: Direct chosen-ciphertext security from identity-based techniques. In: ACM-CCS (2005)
19. Boyle, E., Segev, G., Wichs, D.: Fully leakage-resilient signatures. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 89–108. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_7
20. Camenisch, J., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_8
21. Canetti, R., et al.: Fiat-Shamir: from practice to theory. In: STOC (2019)
22. Canetti, R., Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Adaptive security for threshold cryptosystems. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 98–116. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_7
23. Canetti, R., Goldwasser, S.: An efficient *threshold* public key cryptosystem secure against adaptive chosen ciphertext attack (extended abstract). In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 90–106. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_7
24. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_13
25. Canetti, R., Lombardi, A., Wichs, D.: Fiat-shamir: from practice to theory, Part II (NIZK and correlation intractability from circular-secure FHE). Cryptology ePrint Archive: Report 2018/1248 (2018)

26. Ciampi, M., Parisella, R., Venturi, D.: On adaptive security of delayed-input sigma protocols and Fiat-Shamir NIZKs. In: Galdi, C., Kolesnikov, V. (eds.) SCN 2020. LNCS, vol. 12238, pp. 670–690. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57990-6_33
27. Cramer, R.: Modular design of secure, yet practical cryptographic protocols. Ph.D. thesis, University of Amsterdam (1996)
28. Cramer, R., Damgård, I., Ishai, Y.: Share conversion, pseudorandom secret-sharing and applications to secure computation. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 342–362. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30576-7_19
29. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48658-5_19
30. Cramer, R., Fehr, S.: Optimal black-box secret sharing over arbitrary Abelian groups. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 272–287. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_18
31. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055717
32. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_4
33. Damgård, I.: Efficient concurrent zero-knowledge in the auxiliary string model. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 418–430. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_30
34. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44586-2_9
35. Damgård, I., Thorbek, R.: Linear integer secret sharing and distributed exponentiation. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 75–90. Springer, Heidelberg (2006). https://doi.org/10.1007/11745853_6
36. De Santis, A., Desmedt, Y., Frankel, Y., Yung, M.: How to share a function securely. In: STOC (1994)
37. De Santis, A., Di Crescenzo, G., Ostrovsky, R., Persiano, G., Sahai, A.: Robust non-interactive zero knowledge. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 566–598. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_33
38. Desmedt, Y.: Society and group oriented cryptography: a new concept. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 120–127. Springer, Heidelberg (1988). https://doi.org/10.1007/3-540-48184-2_8
39. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 307–315. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_28
40. Devevey, J., Libert, B., Nguyen, K., Peters, T., Yung, M.: Non-interactive CCA2-secure threshold cryptosystems: achieving adaptive security in the standard model without pairings. Full version, Cryptology ePrint Archive Report (2021)

41. Dodis, Y., Fazio, N.: Public Key trace and revoke scheme secure against adaptive chosen ciphertext attack. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 100–115. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36288-6_8
42. Dodis, Y., Katz, J.: Chosen-ciphertext security of multiple encryption. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 188–209. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30576-7_11
43. Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the fiat-shamir transform. In: Galbraith, S., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 60–79. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34931-7_5
44. Feller, W.: An Introduction to Probability theory and Its Applications. Wiley, New York (1968)
45. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
46. Fouque, P.-A., Pointcheval, D.: Threshold cryptosystems secure against chosen-ciphertext attacks. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 351–368. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_21
47. Frankel, Y., Gemmell, P., MacKenzie, P., Yung, M.: Optimal-resilience proactive public-key cryptosystems. In: FOCS (1997)
48. Frankel, Y., MacKenzie, P., Yung, M.: Adaptively-secure distributed public-key systems. In: Nešetřil, J. (ed.) ESA 1999. LNCS, vol. 1643, pp. 4–27. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48481-7_2
49. Garay, J.A., MacKenzie, P., Yang, K.: Strengthening zero-knowledge protocols using signatures. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 177–194. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_11
50. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC (2009)
51. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC (2008)
52. Goldreich, O.: On (Valiant's) polynomial-size monotone formula for majority. In: Goldreich, O. (ed.) Computational Complexity and Property Testing. LNCS, vol. 12050, pp. 17–23. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43662-9_3
53. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_24
54. Hoory, S., Magen, A., Pitassi, T.: Monotone circuits for the majority function. In: Díaz, J., Jansen, K., Rolim, J.D.P., Zwick, U. (eds.) APPROX/RANDOM -2006. LNCS, vol. 4110, pp. 410–425. Springer, Heidelberg (2006). https://doi.org/10.1007/11830924_38
55. Jarecki, S., Lysyanskaya, A.: Adaptively secure threshold cryptography: introducing concurrency, removing erasures. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 221–242. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_16
56. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 1–20. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_1
57. Katz, J., Vaikuntanathan, V.: Smooth projective hashing and password-based authenticated key exchange from lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 636–652. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_37

58. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_30
59. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_31
60. Libert, B., Nguyen, K., Passelègue, A., Titiu, R.: Simulation-sound arguments for LWE and applications to KDM-CCA2 security. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12491, pp. 128–158. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64837-4_5
61. Libert, B., Nguyen, K., Peters, T., Yung, M.: One-shot fiat-shamir-based NIZK arguments of composite residuosity in the standard model. Cryptology ePrint Archive: Report 2020/1334 (2020)
62. Libert, B., Peters, T., Joye, M., Yung, M.: Non-malleability from malleability: simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 514–532. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_29
63. Libert, B., Stehlé, D., Titiu, R.: Adaptively secure distributed PRFs from LWE. In: TCC (2018)
64. Libert, B., Yung, M.: Adaptively secure non-interactive threshold cryptosystems. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011. LNCS, vol. 6756, pp. 588–600. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22012-8_47
65. Libert, B., Yung, M.: Non-interactive CCA-secure threshold cryptosystems with adaptive security: new framework and constructions. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 75–93. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_5
66. Miao, P., Patel, S., Raykova, M., Seth, K., Yung, M.: Two-sided malicious security for private intersection-sum with cardinality. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020. LNCS, vol. 12172, pp. 3–33. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56877-1_1
67. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41
68. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. SIAM J. Comput. **37**(1), 267–302 (2007)
69. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC (1990)
70. Ostrovsky, R., Yung, M.: How to withstand mobile virus attacks. In: PODC (1991)
71. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
72. Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (Plain) learning with Errors. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11692, pp. 89–114. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_4
73. Rabin, T.: A simplified approach to threshold and proactive RSA. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 89–104. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055722

74. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_35
75. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC (2005)
76. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: FOCS (1999)
77. Shoup, V., Gennaro, R.: Securing threshold cryptosystems against chosen ciphertext attack. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 1–16. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054113
78. Shoup, V., Gennaro, R.: Securing threshold cryptosystems against chosen ciphertext attack. J. Cryptol. **15**(2), 75–96 (2002)
79. Thorbek, R.: Linear integer secret sharing. Ph.D. thesis, Aarhus University (2009)
80. Valiant, L.G.: Short monotone formulae for the majority function, vol. 5, pp. 363–366. Elsevier (1984)
81. Wee, H.: Threshold and revocation cryptosystems via extractable hash proofs. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 589–609. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_32
82. Wee, H.: Dual projective hashing and its applications — lossy trapdoor functions and more. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 246–262. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_16
83. Xie, X., Xue, R., Zhang, R.: Efficient threshold encryption from lossy trapdoor functions. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 163–178. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_11